

ATTY DOCKET #: PR02\_1272A

Due Date: N/A

Confirmation No. 7248

OUR REF: PR02\_1272A/CRW/00703

Applicant: Wataru IKEDA et al.

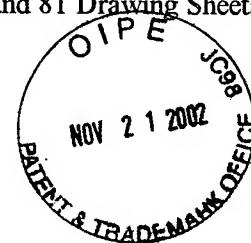
Serial No.: 60/413,153

Filing Date: September 25, 2002

Title: DIGITAL IMAGE CONTENTS RECORDING MEDIUM, APPARATUS AND  
METHOD FOR REPRODUCING THE DIGITAL IMAGE CONTENTS

Receipt of the following papers is acknowledged:

1. Submission of English Translation
2. Verification of Translation
3. Specification, claims and Abstract (144 pages) and 81 Drawing Sheets (A4P, Figs. 1-81)



Date: November 21, 2002

Attorney: CRW/asd

[Check No. N/A ]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of

**Confirmation No. 7248**

Wataru IKEDA et al.

**Docket No. PR02\_1272A**

Serial No. 60/413,153

Filed September 25, 2002

**DIGITAL IMAGE CONTENTS RECORDING  
MEDIUM, APPARATUS AND METHOD FOR  
REPRODUCING THE DIGITAL IMAGE  
CONTENTS**

---

**SUBMISSION OF ENGLISH TRANSLATION WITH STATEMENT OF ACCURACY**

Assistant Commissioner for Patents,  
Washington, D.C.

Sir:

The above-identified U.S. provisional patent application was filed on September 25, 2002 in a language other than English. Therefore, in accordance with 37 CFR 1.78(a)(5)(iv), an English language translation of the provisional application is submitted herewith together with a statement that the translation is accurate.

Respectfully submitted,

Wataru IKEDA et al.

By: \_\_\_\_\_

Charles R. Watts  
Registration No. 33,142  
Attorney for Applicants

CRW/asd  
Washington, D.C. 20006-1021  
Telephone (202) 721-8200  
Facsimile (202) 721-8250  
November 21, 2002

VERIFICATION OF TRANSLATION

I, David Bonnitcha, translator at Nakajima & Matsumura Patent Attorneys Office, 6F Yodogawa 5-Bankan, 3-2-1 Toyosaki, Kita-ku, Osaka, 531-0072, JAPAN, hereby declare that I am conversant with the English and Japanese languages and am a competent translator thereof. I further declare that to the best of my knowledge and belief the following is a true and correct translation made by me of United States Patent Application No. 60/413,153.

Date: 8 November 2002



David Bonnitcha

## TITLE OF THE INVENTION

Digital Image Contents Recording Medium and Apparatus, and  
Method for Reproducing the Digital Image Contents

### 5 BACKGROUND OF THE INVENTION

#### 1. Field of the Invention

The present invention relates to media for recording digital image contents, and an apparatus and a method for reproducing (i.e. "playing back" or simply "playing") the recorded digital image contents.

#### 2. Related Art

The following description relates to a super density digital versatile disc ("SD-DVD" or simply "DVD").

15 Fig.1 shows a structure of an SD-DVD. As shown in the lower half of Fig.1, a logical address space is provided on the DVD between a lead-in and a lead-out. At a head of the logical address space is stored volume information relating to a file system, and following the volume information is 20 stored application data such as video data and audio data.

The file system referred to here conforms to standards such as ISO9660 and UDF (universal disc format), and data on a disc is structured in units known as "directories" and "files". Even with common PCs (personal computers), data

stored in a hard disk is expressed using a directory/file structure based on file systems such as FAT (File Allocation Table) and NTFS (New Technology File System), in order to enhance the usability of the data.

5        In the case of an SD-DVD, a combination of UDF and ISO9660 are used (known as "UDF Bridge"), and a UDF-compatible or ISO9660-compatible file system driver is used to read data from the DVD (note: "DVD" here is a ROM disc used in package media, and cannot be physically 10 written).

      Data stored on the DVD can be viewed, using UDF Bridge, as a directory or a file, as shown in the top-left of Fig.1. Positioned below a "ROOT" directory in Fig.1 is a "VIDEO\_TS" directory in which is stored DVD application data. 15 Application data is stored as a plurality of files, the main files of which include:

VIDEO\_TS.IFO        disc playback control information file  
VTS\_01\_0.IFO        VideoTitleSet#1 playback control  
20                    information file  
VTS\_01\_0.VOB        VideoTitleSet#1 stream file  
.....

      There are two types of extensions: "IFOs", which are

files storing playback control information, and "VOBs", which are files storing MPEG streams relating to AV data. Playback control information includes, for example, (i) information that allows for "interactivity" (technology that 5 allows playback to be varied dynamically in response to user operations) in the DVD being utilized, and (ii) information, such as metadata, for attaching to titles and AV streams. In relation to DVDs, playback control information is generally referred to as "navigation information".

10 Playback control information files include VIDEO\_TS.IFO storing information for managing the entire disc, and VTS\_01\_0.IFO storing playback control information for each video title set (with DVD, it is possible to store a plurality of titles, such as different movies or different 15 versions of the same movie, on a single disc). Here, "01" in the body of the filename shows the number of the video title set. Thus, for example, a VideoTitleSet#2 will be stored in a file having a filename "VTS\_02\_0.IFO".

20 In the right half of Fig.1 is shown a DVD navigation space located in an application layer of the DVD. The DVD navigation space is a logical structural space in which the playback control information mentioned above can be expanded. Information in VIDEO\_TS.IFO is expanded in the DVD navigation space as VMGI (Video Manager Information), whereas playback

control information for VTS\_01\_0.IFO as well as for each of the other video title sets is expanded in the DVD navigation space as VTSI (Video Title Set Information).

VTSI includes PGCI (Program Chain Information) 5 relating to a playback sequence (i.e. "program chain"). PGCI is structured from groups of cells and a type of programming information known as "commands". A cell is itself formed from groups of all or some of the intervals in a VOB ("VOB" = video object; indicates an MPEG stream), and the playback of a cell 10 involves playing back the intervals designated in the VOB by the cell.

Commands are preformed by a virtual machine in the DVD, and are similar to Java Script or the like executed on a browser. However, in comparison to Java Script, which apart 15 from performing logical operations also functions to control windows and browsers (e.g. opening windows on a new browser), the function of a DVD command is limited to logical operations and playback controls relating to AV titles, an example of which is designating a chapter to be played.

20 Cells include, as internal information, the start and end addresses (logical addresses) of VOBs stored on the disc, and a player uses the VOB start and end address information shown in the cells to read and playback data.

Fig.1 is a schematic view of navigation information

embedded in an AV stream. Interactivity, which is a feature of SD-DVDs, is realized not only by the navigation information stored in VIDEO\_TS.IFO, VTS\_01\_0.IFO and other files, but also by a number of pieces of important information that use a dedicated carrier referred to as a navigation pack (abbreviated as "navi-pack" or "NV\_PCK"), and that are multiplexed together with video and audio data in VOBs.

5 Here, a menu will be described to illustrate a simple example of interactivity. On the menu screen appear a number 10 of buttons, each of which shows the type of processing that will be executed if the button is activated. One of the buttons in the menu is already in a selected state (i.e. highlighted by overlaying an opaque color), and the user can operate arrow 15 keys (up/down/left/right) of a remote controller to move the select button to any of the other buttons in the menu. By using the arrow keys of the remote controller to move the highlight to a button that indicates the processing the user wants to be executed, and activating this button (e.g. 20 pressing a RETURN key on the remote controller), a program related to the corresponding command will be executed. Generally, a corresponding title, chapter or the like is played by a command.

The upper-left of Fig.2 summarizes the contents of NV\_PCK.

NV\_PCK includes information such as highlight color information, button information relating to each of the buttons, and the like. The highlight color information shows color pallet information designating the opaque colors of the highlight to be overlaid. The button information shows rectangular area information relating to a position of each of the buttons, move information relating to moving from one button to another (i.e. designating target buttons that correspond to user operations of remote controller arrow keys), and button command information relating to a command to be executed when a button is activated.

As shown in the upper-middle of Fig.2, highlighting in the menu is generated as an overlaid image. The overlaid image is generated by applying one of the colors in the color pallet information to the rectangular information in the button information. The overlaid image is then blended with the background image shown in the right half of Fig.2 for display on a screen.

The realization of a menu in a DVD is as described above. The reason for embedding a part of the navigation data in the stream using the NV\_PCK is to enable the menu information to be dynamically updated in synchronization with the stream, even with respect to applications in which problems relating to sync timing can easily arise (e.g. when a menu is only

displayed for 5 or 10 minutes during the playback of a movie).

Fig.3 relates to VOB images in a DVD. As shown in Fig.3, video, audio, subtitle and other data (see level A in Fig.3) is formed into packets and packs based on the MPEG system 5 (ISO/IEC 13818-1) standard (see B level), and the generated packets and packs are multiplexed to form a single MPEG program stream (see C level). A NV\_PCK that includes button commands for realizing interactivity as described above is also multiplexed together with the generated packets and 10 packs.

A characteristic of multiplexing according to the MPEG system is that even though each piece of multiplexed data is formed into a bit sequence based on the order in which that particular piece of data is to be decoded, it is not 15 always the case that these bit sequences are arranged in the order in which the multiplexed pieces of data (i.e. video, audio, subtitle) will be played back (i.e. decoded). Here, the ordering of the bit sequences is not important because a decode model for decoding MPEG system streams (generally 20 a system target decoder or "STD" as shown in level D of Fig.3) includes decoder buffers corresponding to individual elementary streams, and these decoder buffers function to temporarily store demultiplexed data until a decode timing of the data is reached. The decoder buffers for each type

of elementary stream are different in size. The size of data buffers for video, audio, and subtitle data are 232kB, 4kB, and 52kB, respectively. Because of this difference in data buffer size, the timing of data input into the decoder buffers is different for each of the elementary streams, and thus the order in which the bit sequences are formed in an MPEG system stream does not necessarily match the display (i.e. decoding) timing.

In other words, subtitle data multiplexed next to video data is not necessarily decoded according to the same timing as the video data.

#### **SUMMARY OF THE INVENTION**

As a result of developments in IT technology in recent years, watching a movie on DVD no longer requires the use of a television, and it is now possible, for instance, to watch DVD movies on a PC, or to connect a DVD player to the Internet in the same way as a PC. As a result, the conventional distinction between DVDs as AV devices and PCs as non-AV devices is being made obsolete. Expansions are predicted in the way in which DVD can be enjoyed, such as bringing closer the relationship between the Internet and AV contents stored on DVD.

As shown in Fig.2, at issue here is the use of NV\_PCK

files in MPEG system streams to embed information for realizing interactivity in SD-DVDs.

One of the benefits of the Internet is being able to download new contents from a server, which is commonly conducted using FTP (File Transport Protocol). Enjoying DVD in ways previously unimaginable is now becoming possible, examples of which include downloading new video, audio, subtitle and other data, and using streaming technology to, for example, synchronize playback with data on a disc, or update the playback control information (i.e. interactive scenarios) to enjoy different stories and endings of the same movie.

In view of the above issue, an object of the present invention is to provide HD-DVD technology that realizes new applications for linking disc media and the Internet in ways not possible using conventional DVD technology.

A playback apparatus provided to achieve the above object is for playing back an optical disc that stores at least video data and audio data, the playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing at least one of audio data and subtitle data. Furthermore, the playback apparatus plays back (i) at least one piece of each of the video and audio data stored

on the optical disc in synchronization with (ii) at least one piece of the audio or subtitle data stored in the storage unit.

Here, the storage unit may store sync-playback management information relating to the audio or subtitle data, stored in the storage unit, that is to be played back in synchronization with the video data stored on the optical disc, and the playback apparatus may include units operable to read the sync-playback management information, and perform, in accordance with the sync-playback management information, sync-playback processing of (i) the video data, stored on the optical disc, that is to be synchronously played back, and (ii) the audio or subtitle data, stored in the storage unit, that is to be synchronously played back.

Here, the video and audio data on the optical disc may be stored in a multiplexed state, and (i) a timestamp of the audio data multiplexed on the optical disc with the video data to be synchronously played back may match (ii) a timestamp of the audio data, stored in the storage unit, that is to be synchronously played back. Furthermore, the playback apparatus may include a unit operable to compare (i) the timestamp of the audio data multiplexed on the optical disc with the video data to be synchronously played back with (ii) the timestamp of the audio data, stored in the storage unit,

that is to be synchronously played back.

The above object of the present invention may also be achieved by a playback apparatus for playing back an optical disc that stores at least video data and audio data, the 5 playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing at least video data and audio data. Furthermore, the playback apparatus plays back, in accordance with playback-sequence management information 10 stored in the storage unit, the video and audio data stored on the optical disc, and the video and audio data stored in the storage unit.

The above object of the present invention may also be achieved by a playback apparatus for playing back an optical 15 disc that stores at least video data and audio data, the playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing at least video data and audio data. Furthermore, the video and audio data stored on 20 the optical disc includes, when playback is to be conducted by the playback apparatus, angle intervals for dynamically switching playback scenes, and the playback apparatus plays back, as one of the angle intervals, at least one piece of the video data stored in the storage unit.

Here, the optical disc may store first playback management information for regulating a playback sequence of the video and audio data stored on the optical disc, and the storage unit may store second playback management information for regulating a playback sequence of (i) the video and audio data stored on the optical disc and (ii) the video and audio data stored in the storage unit. Furthermore, the playback apparatus may generate third playback management information based on the first and second playback management information.

Here, a part or all of the first playback management information having the same identification information as the second playback management information may be nullified by the second playback management information.

Here, the storage unit may store, for each optical disc, (i) audio or subtitle data to be played back in synchronization with video and audio data stored on the optical disc, and (ii) video and audio data to be played back, in a predetermined playback sequence, with video and audio data stored on the optical disc. Furthermore, the playback apparatus may store a table for managing, for each optical disc, (i) the audio or subtitle data, stored in the storage unit, that is to be synchronously played back, and (ii) the video and audio data, stored in the storage unit, that is

to be played back in the predetermined playback sequence.

Here, the playback apparatus may be connected to a network to which another playback apparatus is connected, and the other playback apparatus may refer, at a time of 5 playing back an optical disc, to the table stored in the playback apparatus.

Here, the playback apparatus may include an interface for connecting to the Internet, and may obtain, from the Internet, audio data to be played back in synchronization 10 with the video data stored on the optical disc.

The above object of the present invention may also be achieved by a playback apparatus for playing back an optical disc that stores at least video data and audio data, the playback apparatus including a removable nonvolatile memory, 15 and the removable nonvolatile memory storing playback control information for managing the playback of the video and audio data stored on the optical disc. Furthermore, the playback apparatus reads the playback control information from the removable nonvolatile memory, and plays back, in 20 accordance with the playback control information, the video and audio data stored on the optical disc.

The above object of the present invention may also be achieved by a playback method in a playback system for playing back an optical disc that stores at least video data and audio

data, the playback system including a storage unit that is at least one of a magnetic storage unit and a removable nonvolatile memory, and the storage unit storing at least one of audio data and subtitle data. Furthermore, the 5 playback method includes the step of playing back (i) at least one piece of each of the video and audio data stored on the optical disc in synchronization with (ii) at least one piece of the audio or subtitle data stored in the storage unit.

Here, the storage unit may store sync-playback 10 management information relating to the audio or subtitle data, stored in the storage unit, that is to be played back in synchronization with the video data stored on the optical disc, and the playback method may include the steps of reading the sync-playback management information, and performing, 15 in accordance with the sync-playback management information, sync-playback processing of (i) the video data, stored on the optical disc, that is to be synchronously played back, and (ii) the audio or subtitle data, stored in the storage unit, that is to be synchronously played back.

20 Here, the video and audio data on the optical disc may be stored in a multiplexed state, and (i) a timestamp of the audio data multiplexed on the optical disc with the video data to be synchronously played back may match (ii) a timestamp of the audio data, stored in the storage unit, that

is to be synchronously played back. Furthermore, the playback method may include the step of comparing (i) the timestamp of the audio data multiplexed on the optical disc with the video data to be synchronously played back with (ii) the 5 timestamp of the audio data, stored in the storage unit, that is to be synchronously played back.

The above object of the present invention may also be achieved by a playback method in a playback system for playing back an optical disc that stores at least video data and audio 10 data, the playback system including a storage unit that is at least one of a magnetic storage unit and a removable nonvolatile memory, and the storage unit storing at least video data and audio data. Furthermore, the playback method includes the step of playing back, in accordance with 15 playback-sequence management information stored in the storage unit, the video and audio data stored on the optical disc and the video and audio data stored in the storage unit.

The above object of the present invention may also be achieved by a playback method in a playback system for playing 20 back an optical disc that stores at least video data and audio data, the playback system including a storage unit that is at least one of a magnetic storage unit and a removable nonvolatile memory, and the storage unit storing at least video data and audio data. Furthermore, the video and audio

data stored on the optical disc includes, when playback is to be conducted by the playback system, angle intervals for dynamically switching playback scenes, and the playback method includes the step of playing back, as one of the angle intervals, at least one piece of the video data stored in the storage unit.

Here, the optical disc may store first playback management information for regulating a playback sequence of the video and audio data stored on the optical disc, and the storage unit may store second playback management information for regulating a playback sequence of (i) the video and audio data stored on the optical disc and (ii) the video and audio data stored in the storage unit. Furthermore, the playback method may include the step of generating third playback management information based on the first and second playback management information.

Here, a part or all of the first playback management information having the same identification information as the second playback management information may be nullified by the second playback management information.

Here, the playback method may also include the step of storing in the storage unit, for each optical disc, (i) audio or subtitle data to be played back in synchronization with video and audio data stored on the optical disc, and (ii)

video and audio data to be played back, in a predetermined playback sequence, with video and audio data stored on the optical disc. Furthermore, the playback method may include the step of storing, in the playback system, a table for 5 managing, for each optical disc, (i) the audio or subtitle data, stored in the storage unit, that is to be synchronously played back, and (ii) the video and audio data, stored in the storage unit, that is to be played back in the predetermined playback sequence.

10 Here, the playback system may be connected to a network to which another playback system is connected, and the other playback system may refer, at a time of playing back an optical disc, to the table stored in the playback system.

15 Here, the playback system may include an interface for connecting to the Internet, and the playback method may include the step of obtaining, from the Internet, audio data to be played back in synchronization with the video data stored on the optical disc.

20 The above object of the present invention may also be achieved by a playback method in a playback system for playing back an optical disc that stores at least video data and audio data, the playback system including a removable nonvolatile memory, and the removable nonvolatile memory storing playback control information for managing the playback of

the video and audio data stored on the optical disc. Furthermore, the playback method includes the steps of reading the playback control information from the removable nonvolatile memory, and playing back, in accordance with the 5 playback control information, the video and audio data stored on the optical disc.

The above object of the present invention may also be achieved by an optical disc storing at least video data and audio data, and played back by a playback apparatus, the 10 playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing at least one of audio data and subtitle data. Furthermore, at least one piece of the video data stored on the optical disc is played back in 15 synchronization with at least one piece of the audio or subtitle data stored in the storage unit, and the optical disc stores a part or all of reference information that is for specifying the audio or subtitle data, stored in the storage unit, that is to be played back in synchronization 20 with the video data stored on the optical disc.

Here, the storage unit may store sync-playback management information relating to the audio or subtitle data, stored in the storage unit, that is to be played back in synchronization with the video data stored in the optical

disc. Furthermore, the video data to be synchronously played back and the audio or subtitle data to be synchronously played back may be sync-playback processed by the playback apparatus in the accordance with the sync-playback management 5 information.

Here, the video and audio data on the optical disc may be stored in a multiplexed state, and (i) a timestamp of the audio data multiplexed on the optical disc with the video data to be synchronously played back may match (ii) a 10 timestamp of the audio data, stored in the storage unit, that is to be synchronously played back.

Here, the optical disc and the storage unit may each store access-support information relating to video and audio data stored respectively, and the access-support information 15 may be a table for managing, in arbitrary units, time information and data size information. Furthermore, (i) an audio playback time period, managed in the table on the optical disc, that relates to the audio data stored on the optical disc with video data that is to be synchronously 20 played back may match (ii) an audio playback time period, managed in the table in the storage unit, that relates to the audio data, stored in the storage unit, that is to be synchronously played back.

The above object of the present invention may also be

achieved by an optical disc storing at least video data and audio data, and played back by a playback apparatus, the playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing video data. Furthermore,  
5 the video data stored on the optical disc may be played back, in a predetermined playback sequence, with the video data stored in the storage unit, and the optical disc may store a part or all of reference information that is for specifying  
10 the video data, stored in the storage unit, that is to be played back, in the predetermined playback sequence, with the video data stored on the optical disc.

The above object of the present invention may also be achieved by an optical disc storing at least video data and audio data, and played back by a playback apparatus, the playback apparatus including a storage unit that is at least one of a magnetic storage device and a removable nonvolatile memory, and the storage unit storing at least video data and audio data. Furthermore, the video and audio data stored on  
15 the optical disc may include, when playback is to be conducted by the playback apparatus, multi-angle intervals for dynamically switching playback scenes, and at least one piece of each of the video and audio data stored in the storage unit may be played back as one of the angles of the multi-angle  
20

intervals. Furthermore, the optical disc may store a part or all of reference information that is for specifying the video and audio data, stored in the storage unit, that is to be played back as one of the angles of the multi-angle 5 intervals.

Here, the playback apparatus may include an interface for connecting to the Internet, and may obtain, from the Internet, audio data to be played back in synchronization with the video data stored on the optical disc, and the optical 10 disc may store access point information relating to the Internet.

#### BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects, advantages and features of the 15 invention will become apparent from the following description thereof taken in conjunction with the accompanying drawings that illustrate specific embodiments of the present invention.

20 In the drawings:

Fig.1 is a structural diagram of a DVD;

Fig.2 is a structural diagram of a highlight;

Fig.3 shows an example of multiplexing in a DVD;

Fig.4 shows a data hierarchy in an HD-DVD;

Fig.5 is a structural diagram of a logical space in an HD-DVD;  
Fig.6 is a schematic block diagram of an HD-DVD player;  
Fig.7 is a structural block diagram of the HD-DVD player;  
Fig.8 shows an application space in an HD-DVD;  
5 Fig.9 is a structural diagram of MPEG streams (VOBs);  
Fig.10 is a structural diagram of a pack;  
Fig.11 shows a relationship between an AV stream and a player  
structure;  
Fig.12 is a model diagram of the continuous supply of AV data  
10 to a track buffer;  
Fig.13 is a structural diagram of a VOB information file;  
Fig.14 shows a time map;  
Fig.15 shows a method of obtaining address information using  
a time map;  
15 Fig.16 is a structural diagram of a playlist file;  
Fig.17 is a structural diagram of a program file that  
corresponds to the playlist;  
Fig.18 is a structural diagram of a management information  
file relating to an entire BD disc;  
20 Fig.19 is a structural diagram of a file storing a global  
event handler;  
Fig.20 shows an exemplary time event;  
Fig.21 shows an exemplary user event;  
Fig.22 shows an exemplary global event handler;

Fig.23 is a structural diagram of a virtual machine;  
Fig.24 shows a player variable table;  
Fig.25 shows an exemplary event handler (time event);  
Fig.26 shows an exemplary event handler (user event);  
5 Fig.27 is a flowchart of basic processing conducted in a  
player;  
Fig.28 is a flowchart of playlist playback processing;  
Fig.29 is a flowchart of event processing;  
Fig.30 is a flowchart of subtitle processing;  
10 Fig.31 is a conceptual diagram of a virtual package;  
Fig.32 shows exemplary virtual package applications;  
Fig.33 shows an exemplary "language credit" in a virtual  
package;  
Fig.34 shows an exemplary "parental lock" in a virtual  
15 package;  
Fig.35 shows an exemplary "director's cut" in a virtual  
package;  
Fig.36 shows an exemplary "audio supplement" in a virtual  
package;  
20 Fig.37 shows an exemplary "multi-angle" in a virtual package;  
Fig.38 is a structural diagram of directories and files on  
a storage disc;  
Fig.39 shows the generation of a virtual package;  
Fig.40 is a structural diagram of directories and files on

a storage disc;

Fig.41 is a structural diagram of a management information file relating to an entire virtual package;

Fig.42 is a structural diagram of a playlist file in a virtual package;

5

Fig.43 is a structural diagram of a VOB information file in a virtual package;

Fig.44 shows the offset between VOBs using an SCR base;

Fig.45 shows a time map generation rule in an embodiment 2;

10 Fig.46 is a schematic block diagram of a player for playing virtual packages;

Fig.47 is a functional block diagram of AV data processing in a player for playing virtual packages;

15 Fig.48 shows a detailed structure of a demultiplexer and an audio processor;

Fig.49 is a flowchart of processing conducted prior to playback of a virtual package;

Fig.50 is a flowchart of basic processing related to a virtual package;

20 Fig.51 is a flowchart of playlist playback processing related to a virtual package;

Fig.52 is a flowchart of AV playback processing related to a virtual package;

Fig.53 is a flowchart of first VOB sync-playback processing

related to a virtual package;

Fig.54 is a flowchart of second VOB sync-playback processing  
related to a virtual package;

Fig.55 is a flowchart of event processing performed during  
5 the playlist playback of a virtual package;

Fig.56 is a flowchart of subtitle switching processing  
performed during the playlist playback of a virtual  
package;

Fig.57 shows a screen for making a playback selection from  
10 a plurality of virtual packages;

Fig.58 is a schematic block diagram of a player for making  
a playback selection from a plurality of virtual  
packages;

Fig.59 is a variable table in a player for playing back virtual  
15 packages;

Fig.60 is a structural diagram of virtual package management  
information;

Fig.61 is a flowchart of processing conducted prior to and  
including the selection of a virtual package;

20 Fig.62 is a schematic block diagram of a player for playing  
back virtual packages in an embodiment 4;

Fig.63 is a structural diagram of virtual package table  
information;

Fig.64 is a flowchart of processing conducted prior to

starting playback of a virtual package in embodiment  
4;

Fig.65 is a flowchart of processing to start playback of a  
virtual package in embodiment 4;

5 Fig.66 shows exemplary structures of a player in embodiment  
4;

Fig.67 is a flowchart of processing conducted prior to  
starting the playback of a virtual package in players  
having various structures;

10 Fig.68 is a structural diagram of a multi-angle;

Fig.69 is a structural diagram of playlist information in  
an embodiment 5;

Fig.70 shows an address management method for an interleaved  
block;

15 Fig.71 is a structural diagram of VOB management information  
in embodiment 5;

Fig.72 is a flowchart of event processing in embodiment 5;

Fig.73 is a flowchart of VOB playback processing in  
embodiment 5;

20 Fig.74 is a functional block diagram of AV data processing  
in a player for playing back virtual packages in an  
embodiment 6;

Fig.75 is a structural diagram of VOB information in  
embodiment 6;

Fig.76 is a structural diagram of playlist information in embodiment 6;

Fig.77 is a flowchart of AV playback processing related to a virtual package in embodiment 6;

5 Fig.78 is a flowchart of first VOB sync-playback processing related to a virtual package in embodiment 6;

Fig.79 is a flowchart of second VOB sync-playback processing related to a virtual package in embodiment 6;

10 Fig.80 shows exemplary virtual package applications using a DVD; and

Fig.81 is a structural diagram of DVD and BD combined.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

##### Embodiment 1

15 An embodiment 1 of the present invention will be now described.

##### *Logical Data Structure on Disc*

Fig.4 shows a structure of an HD-DVD (hereafter, 20 "Blu-ray disc" or simply "BD"). In particular, Fig.4 shows a structure of a BD disc 104 (i.e. a disc medium) and data stored on the disc. The data stored on BD disc 104 includes AV data 103, BD management information 102 (including AV playback sequences and management information relating to the AV data),

and a BD playback program 101 for realizing interactivity. Although the description in the present embodiment focuses on a BD disc that realizes an AV application for playing back movies and other AV contents, it is of course possible to 5 apply the BD disc, like a CD-ROM or a DVD-ROM, as a storage medium for use with a computer.

Fig.5 shows logical data stored on the above BD disc. As with other optical discs such as DVDs or CDs, for example, the BD disc has a spiral-shaped storage area that spirals 10 from the inner circumference outwards to the outer circumference of the disc, and in between a lead-in around the inner circumference and a lead-out around the outer circumference is provided a logical address space for storing logical data. On the inner side of the lead-in is a special 15 area, referred to as a BCA (burst cutting area), which can only be read by a driver. Since this area cannot be read from an application, it is often used, for example, in copyright protection technology.

At the start of the logical address space is stored file 20 system information (volume information), after which is stored video data and other application data.

As described in the above related art, the file system conforms to standards such as UDF, ISO9660, and the like, and similar to a conventional PC, the file system allows for

stored logical data to be read using a directory/file structure.

In the present embodiment, the directories and files on the BD disc are structured with a ROOT directory at the 5 top, and a BD VIDEO directory positioned directory below the ROOT directory. The BD VIDEO directory stores AV contents, management information, and other data (i.e. 101, 102, 103 in Fig.4) handled by the HD-DVD.

In the BD VIDEO directory are stored the following seven 10 types of files.

- (1) BD.INFO (fixed filename): file storing information (BD management information) that relates to the entire BD disc; BD player reads this file first
- (2) BD.PROG (fixed filename): file storing a program (BD playback program) that relates to the entire BD disc 15
- (3) XXX.PL ("XXX" variable; extension "PL" fixed): file storing playlist information (BD management information) that relates to scenarios; one file per playlist
- (4) XXX.PROG ("XXX" variable; extension "PROG" fixed): file storing a program (BD playback program); one program per playlist; correspondence with a particular playlist is identified in the body of the filename (i.e. 20 the "XXX" part of the filenames match)

(5) YYY.VOB ("YYY" variable; extension "VOB" fixed): file storing a VOB (AV data) as described in the related art; one file per VOB

5 (6) YYY.VOBI ("YYY" variable; extension "VOBI" fixed): file storing management information (BD management information) relating to a VOB; one file per VOB; correspondence with a particular VOB is identified in the body of the filename (i.e. the "YYY" part of the filenames match)

10 (7) ZZZ.PNG ("ZZZ" variable; extension "PNG" fixed): file storing AV data (PNG image data) for structuring subtitles and menus ("PNG": image file format standardized by W3C, and pronounced "ping"); one file per PNG image

15

#### *Player Structure*

Next, a structure of a player for playing back the above BD disc will be described with reference to Figs. 6 and 7.

Fig. 6 is a block diagram showing a general functional 20 structure of the player.

Data stored on a BD disc 201 is read via an optical pickup 202. Read data is stored in a dedicated memory in accordance with the data type. BD playback programs (i.e. contents of BD.PROG and XXX.PROG) are stored in a program storage memory

203, BD management information (i.e. contents of BD.INFO, XXX.PL, YYY.VOBI) is stored in a management information storage memory 204, and AV data (i.e. contents of YYY.VOB, ZZZ.PNG) is stored in an AV storage memory 205.

5 BD playback programs stored in program storage memory 203 are processed by a program processing unit 206, BD management information stored in management information storage memory 204 is processed by a management information processing unit 207, and AV data stored in AV data storage 10 memory 205 is processed by a presentation processing unit 208.

Program processing unit 206 (i) receives from management information processing unit 207, playlist information (relating to playlists to be played), event 15 information (such as that relating to an execution timing of programs) and the like, and (ii) conducts program processing. Furthermore, it is possible in a program to dynamically change the playlist to be played back, and this is realized by sending a playlist playback command to 20 management information processing unit 207. Program processing unit 206 receives an event from a user (i.e. a request from a remote controller), and executes processing when there exists a program corresponding to the user event.

Management information processing unit 207 receives

instructions from program processing unit 206, analyzes the corresponding playlist and VOB management information corresponding to the playlist, and instructs presentation processing unit 208 to playback target AV data. Furthermore, 5 unit 207 receives reference time information from unit 208, instructs unit 208 to terminate playback of the AV data, and generates an event showing a program execution timing with respect to unit 206.

Presentation processing unit 208 includes separate 10 decoders for each of the three data types (i.e. video, audio, subtitle/image), and conducts decoding and outputting of AV data in accordance with instructions from management information processing unit 207. After being decoded, video and subtitle/image data is drawn in dedicated planes; that 15 is, a video plane 210 and an image plane 209. Integration processing of the video is then conducted by integration processing unit 211, and the integrated video is outputted to a television or similar display device.

As shown in Fig. 6, a structure of the BD player is based 20 on the structure of data (see Fig. 4) stored on the BD disc.

Fig. 7 is a block diagram showing a detailed structure of the above player. The following units in Fig. 7 either correspond to or are an extension of units in Fig. 6: an image memory 308 and a track buffer 309 in Fig. 7 with AV storage

memory 205 in Fig. 6; a program processor 302 and UOP manager 303 with program processing unit 206; a scenario processor 305 and a presentation controller 306 with management information processing unit 207; and a clock 307, a 5 demultiplexer 310, an image processor 311, a video processor 312, and an audio processor 313 with presentation processing unit 208.

VOB data (MPEG stream) and image data (PNG) read from BD disc 201 is stored in track buffer 309 and image memory 10 308, respectively. Based on the time shown by clock 307, demultiplexer 310 extracts VOB data stored in track buffer 309, and sends video data to video processor 312 and audio data to audio processor 313. As determined by the MPEG system standard, video processor 312 and audio processor 313 are 15 each structured from decoder buffers and decoders. Thus, video and audio data sent from demultiplexer 310 to video processor 312 and audio processor 313, respectively, is temporarily stored in the respective decoder buffers of these two units, and then decoded by the respective decoders in 20 accordance with clock 307.

The following two methods of processing PNGs stored in image memory 308 are possible.

In the case of image data for a subtitle, an instruction relating to a decode timing is issued by presentation

controller 306. Scenario processor 305 receives time information from clock 307, and in order to appropriately display the subtitle, scenario processor 305 instructs presentation controller 306 to start or stop the display of the subtitle when a subtitle display time (i.e. display start or end time) is reached. On receipt of a decode/display instruction from presentation controller 306, image processor 311 extracts the corresponding PNG data from image memory 308, decodes the extracted PNG data, and draws the decoded PNG data in image plane 314.

On the other hand, in the case of image data for a menu, an instruction relating to a decode timing is issued by program processor 302. The decode timing instructed by program processor 302 varies depending on the BD program that program processor 302 is processing, and an overall value is not determined.

As shown in Fig.6, image data and video data is stored in image plane 314 and video plane 315, respectively, after being decoded, and then integrated and outputted by integration processing unit 316.

Management information (scenario, AV management information) read from BD disc 201 is stored in management information storage memory 304, and scenario information (BD.INFO, XXX.PL) is read and processed by scenario processor

305. AV management information (YYY.VOBI) is read and processed by presentation controller 306.

Scenario processor 305 analyzes information for a playlist, and instructs presentation controller 306 in relation to (i) the VOB referred to by the playlist and (ii) a playback positioning of the VOB. Presentation controller 306 analyzes management information relating to the targeted VOB (YYY.VOBI), and instructs drive controller 317 to read the targeted VOB.

As instructed by presentation controller 306, drive controller 317 moves the optical pickup, and reads the targeted AV data. As mentioned above, the read AV data is stored in image memory 308 or track buffer 309.

Scenario processor 305 monitors the time of clock 307, and sends events to program processor 302 based on a timing configured in the management information.

BD programs (BD.PROG or XXX.PROG) stored in program storage memory 301 are executed by program processor 302. The processing of BD programs by program processor 302 is conducted when an event is sent from either scenario processor 305 or UOP manager 303. In the case of a user request being received from a remote controller, UOP manager 303 generates an event in program processor 302.

### *Application Space*

Fig. 8 shows an application space in an HD-DVD.

Playlists are one of the playback units in the application space of an HD-DVD. A playlist includes a static scenario structured from the playback sequences of cells, and a dynamic scenario described by a program. If there is no dynamic scenario, the playback of a playlist simply involves playing back the individual cells in order, and playback of the playlist is completed when all the cells have been played back. On the other hand, it is possible for a program to dynamically alter (i) the playback description to include playlists other than the current playlist, and (ii) the playback target in accordance with a user selection or a player status. A menu is illustrated here as a typical example. In the case of an HD-DVD, a menu refers to the dynamic selection of scenarios (i.e. playlists) for playback by a user selection.

A "program" here refers to an event handler executed by a time event or a user event.

A time event is generated on the basis of time information embedded in a playlist. The event sent from scenario processor 305 to program processor 302 as shown in Fig. 7 corresponds to a time event. When a time event is issued, program processor 302 executes an event handler identified

by an ID. As described above, the executed program is able to instruct the playback of other playlists. In this case, playback of the playlist currently being reproduced is ended, and program processor 302 proceeds to playback the designated 5 playlist.

User events are generated by a user operation of a remote controller. There are two main types of user events. One type is a "menu selection event", which is generated by the operation of an arrow key (up/down/left/right), RETURN 10 key or the like of a remote controller. Event handlers corresponding to menu selection events are only valid for a limited period shown in the playlist (the valid period of each event handler is configured in information relating to the playlist). When an arrow key (up/down/left/right), a 15 RETURN key or the like of a remote controller is operated (e.g. pressed), a search is conducted for a valid event handler, and if a valid event handler is located, the located event handler is executed. If a valid event handler is not located, the menu selection event is ignored.

20 The second type of user event is a "menu call event", which is generated by the operation of a MENU key. When a menu call event is generated, a global even handler is called. A global even handler is not dependant on a playlist and is always valid. This function allows menu call in a DVD to be

realized (i.e. during playback of a title, an audio or subtitle menu, etc can be called, and after conversion of the audio or subtitle, playback of the title is resumed from where it was interrupted).

5 Cells, which are the units structuring a static scenario in a playlist, indicate all or some of the playback interval in a VOB (MPEG stream). A cell includes the playback intervals in a VOB as start and end time information. VOB management information (VOBI), which corresponds one-to-one 10 with the VOBS, includes a time map (TM), and by referring to this time map it is possible to locate the read start addresses and end addresses in the VOB from the above VOB playback start and end times. A detailed description of time maps is given in a later section.

15

#### *Detailed Description of VOB*

Fig.9 is a structural diagram of MPEG streams (VOBs) used in the present embodiment.

As shown in Fig.9, a VOB is structured by a plurality 20 of VOBUs (video object units). VOBUs are based on what are referred to in MPEG video streams as GOPs (group of pictures), and are one of the playback units in a multiplexed stream that also includes audio data. A VOBU may have a playback time period of 0.4 seconds to 1.0 seconds, and a playback

time period of 0.5 seconds is common. This is derived from the fact that a GOP in an MPEG stream is structured from 15 frames per second (according to the NTSC standard).

A VOBU includes a video pack ("V\_PCK") and an audio pack 5 ("A\_PCK"). Each pack is structured by a single sector, and is a 2kB unit according to the present embodiment.

Fig.10 shows a structure of a pack.

As shown in Fig.10, elementary data, which includes video data and audio data, is stored sequentially starting 10 at a head of a packet data storage area known as a "payload". A packet header is attached to the payload to form a single packet. The packet header stores information showing the type of stream (i.e. video or audio) in which the data is included. If there is a plurality of video streams or audio streams, 15 the packet header stores an ID ("stream\_id") identifying the stream in which the data is included. The packet header also stores a DTS (decoding timestamp) and a PTS (presentation timestamp), which is information relating to decoding and display times. The DTS/PTS are not necessarily recorded in 20 all packet headers, and the rules for recording these timestamps are regulated by MPEG. A detailed description of these rules is contained in the MPEG system (ISO/IEC 13818-1) standards, and is not included here.

Furthermore, a header (pack header) is attached to a

packet to form a pack. The pack header stores an SCR (system clock reference) timestamp showing when the pack is to pass through a demultiplexer and be inputted into a decoder buffer corresponding to the elementary stream type.

5

#### *VOB Interleaved Storage*

The interleaved storage of VOB files will now be described with reference to Figs.11 and 12.

10 The upper half of Fig.11 shows a partial structure of the player described above. As shown in Fig.11, with VOBs (i.e. MPEG streams), data on a BD disc is inputted to a track buffer via an optical pickup, and with PNGs (i.e. image data), data on the BD disc is inputted to an image memory.

15 The track buffer is a FIFO, and the inputted VOB data is sent to a demultiplexer in the order in which it is inputted. Here, each pack is removed from the track buffer in accordance with an SCR timestamp as described above, and the data is sent via the demultiplexer to a video processor or an audio processor. With image data, on the other hand, a presentation 20 controller instructs as to which image is to be drawn. Furthermore, if the image data to be drawn is for a subtitle, the image data is erased from the image memory after being drawn. On the other hand, if the image data to be drawn is for a menu, the image data is left in the image memory. This

is because the drawing of a menu is partially dependant on user operations, and there exists the possibility that the same image will be drawn a plurality of times.

The lower half of Fig.11 shows the interleaved storage of VOB files and PNG files on the BD disc. Generally, in the case of a ROM (e.g. CD-ROM, DVD-ROM, etc), AV data, which forms a series of continuous playback units, is stored continuously. If data is stored continuously, the driver is only required to read the data sequentially and send the read data to the player. However, if the AV data is divided up and scattered around the disc, a seek operation is executed between each of the continuous intervals, and during the execution of the seek operation, reading of data is stopped, and this may also result in the supply of data being stopped.

As with a ROM, VOB files on a HD-DVD are preferably stored in a continuous area, although with data such as subtitle data, for example, that is played back in synchronization with video data, a method of reading the subtitle data from the BD disc in relation to the VOB files is required.

One method of reading subtitle data is to read all the image data for a subtitle (PNG files) prior to a start of the VOB playback. However, this method requires a large memory for temporary storage, and is not practical.

In comparison, the present embodiment employs a method

in which the VOB files are divided into a number of blocks, and the image data is stored as interleaves between the blocks of VOB files. The lower half of Fig.11 shows an example of this interleaved storage.

5 By appropriately interleaving the image data between the VOB file blocks, it is possible to store image data in the image memory at the required timing, without the need for a large temporary storage memory. However, the reading of VOB data is still interrupted, of course, while image data 10 is being read.

Fig.12 shows a VOB data continuous supply model that uses a track buffer to resolves this problem.

As described above, the VOB data is temporarily stored in a track buffer, and by providing different rates of data 15 input and data output to and from the track buffer, the amount of data stored in the track buffer will increase (or decrease) for the duration that data is read from the BD disc.

Here, let  $V_a$  be the data input rate to the track buffer, and  $V_b$  be the data output rate from the track buffer. As shown 20 in the upper half of Fig.12, the continuous storage area of the VOB spans from logical addresses  $a_1$  to  $a_2$ . Image data is stored between logical addresses  $a_2$  to  $a_3$ , and during this interval it is not possible to read VOB data.

The lower half of Fig.12 represents the content of the

track buffer. The horizontal axis shows time and the vertical axis shows the amount of data stored in the track buffer. At time  $t_1$  the reading of  $a_1$  (i.e. start of VOB continuous storage area) is commenced. From time  $t_1$  onwards, data is stored in the track buffer at a rate  $V_a - V_b$ . Needless to say, this rate is the difference between the data input and output rates. Time  $t_2$  is the reading-in time of  $a_2$  data,  $a_2$  being the end point of the VOB continuous storage area. In other words, between times  $t_1$  and  $t_2$ , the data amount in the track buffer increases at the rate  $V_a - V_b$ , and the data storage amount at time  $t_2$  can be expressed by the following formula:

$$\text{Formula 1} \quad B(t_2) = (V_a - V_b) \times (t_2 - t_1)$$

15 Image data is stored from addresses  $a_2$  to  $a_3$ , and thus the input to the track buffer is reduced to zero, and the data amount in the track buffer decreases at an output rate  $-V_b$ . This continues until the reading position  $a_3$  (corresponding to time  $t_3$ ) is reached.

20 Here, it is important to note that if the data amount stored in the track buffer reaches zero before time  $t_3$ , the supply of VOB data to the decoder will run out, and this may result in the VOB playback being terminated. In other words, as long as there is data in the track buffer at time  $t_3$ ,

playback of the VOB will continue uninterrupted.

This condition can be expressed by the following formula:

5                    Formula 2                     $B(t2) \geq -vb \times (t3 - t2)$

In other word, the image data is preferably arranged to satisfy Formula 2.

10   *Navigation Data Structure*

The structure of navigation data (BD management data) on an HD-DVD will now be described with reference to Figs.13 to 19.

15   Fig.13 shows the content of a VOB management information file (YYY.VOBI).

VOB management information includes VOB stream attribute information ("Attribute") and a time map ("TMAP"). The stream attribute is structured to include a plurality of video attributes ("Video") and audio attributes (Audio#0 ~ Audio#m). In the case of an audio stream, in particular, it is possible for a VOB to include a plurality of audio streams at the same time, and thus the existence of a data field is indicated by information showing the number of audio streams ("Number").

Fields included in the video attribute (“Video”), as well as the various values that these fields may take are as follows:

5           Compression Format (“Coding”):

    MPEG1  
    MPEG2  
    MPEG4

10           Resolution (“Resolution”):

    1920 × 1080  
    1280 × 720  
    720 × 480  
    720 × 565

15

         Aspect Ratio (“Aspect”):

    4 : 3  
    16 : 9

20           Frame Rate (“Framerate”):

    60  
    59.94  
    50  
    30

29.97

25

24

5 Fields included in the audio attribute ("Audio"), as well as the various values these fields may take are as follows:

Compression Format ("Coding"):

10 AC3

MPEG1

MPEG2

LPCM

15 Number of Channels ("Ch"):

1~8

Language Attribute ("Language"):

20 A time map (TMAP) is a table that has information relating to each VOBU. This information includes the number of VOBUS ("Number") in the VOB, and VOBU information (VOBU#1~VOBU#n). Each piece of VOBU information includes a playback time period of the VOBU ("Duration") and a data size

of the VOBU ("Size").

Fig.14 shows a detailed structure of VOBU information.

As commonly known, a MPEG stream has two facets: time and data size. For example, AC3, which is audio compression standard, conducts constant bit rate compression, and thus the time-address relationship can be derived using Formula 1. However, in the case of MPEG video data, each frame has a constant display time period (e.g. 1/29.97 secs/frame according to NTSC), and a data size of each frame after compression varies greatly depending on factors such as the picture attribute and picture type used in compression (i.e. I/P/B picture). Consequently, in the case of MPEG video, the time-address relationship cannot be expressed using a general formula.

Naturally, it is also not possible to use a general formula to express the time-data relationship of an MPEG system stream (i.e. VOB) multiplexed with MPEG video data. Instead of a formula, the time-address relationship of a VOB can be expressed using a time map (TMAP). As shown in Fig.14, a time map is a table that includes, for each VOBU, the number of frames and packs in the VOBU as entries.

The use of a time map will now be described with reference to Fig.15.

As shown in Fig.15, when time information is provided,

a search is firstly conducted to locate the VOBU to which the provided time belongs. Here, the number of frames in each of the VOBUs is calculated, and the VOBU whose total number of frames is greater than or equal to the provided time (i.e. 5 converted into an equivalent number of frames) is determined as the VOBU to which the time belongs. Next, the size of all of the VOBUs in the time map prior to the located VOBU are added together, and the calculated value is determined as the start address of the pack to be read in order to playback 10 frames that include the provided time.

Next, an internal structure of playlist information ("XXX.PL") will be described with reference to Fig.16.

Playlist information is structured from cell lists and event lists.

15 A cell list ("CellList") is a playback sequence in the playlist, and the cells are played back in the order in which they are included in the list. The cell list includes the number of cells ("Number"), and cell information (Cell#1~Cell#n).

20 Cell information (Cell#) includes a VOB filename ("VOBName"), a valid interval start time ("In") of the VOB, a valid interval end time ("Out") of the VOB, and a subtitle table ("SubtitleTable"). The valid interval start and end times (In, Out) are expressed using VOB frame numbers, and

the address of VOB data for playback can be obtained using the above time map (TMAP).

The subtitle table (SubtitleTable) includes subtitle information to be played back in synchronization with the 5 VOB. Like audio, a subtitle can have a plurality of languages. The first piece of information in the subtitle table is the number of languages ("Number"), which is followed by tables of each of the languages (Language#1~Language#k).

Each language table (Language#) is structured from 10 language information ("Language"), the number of pieces of subtitle information relating to subtitles to be displayed ("Number"), and subtitle information relating to subtitles to be displayed (Speech#1~Speech#j). The speech information (Speech#) is structured from a corresponding image data 15 filename ("Name"), a subtitle display start time ("In"), a subtitle display end time ("Out"), and a subtitle display position ("Position").

The event list ("EventList") is a table defining the events given within the playlist. The event list is 20 structured from the number of events ("Number"), after which follows the individual events (Event#1~Event#m). The events (Event#) are structured from an event type ("Type"), an event ID ("ID"), an event generation time ("Time"), and a valid period ("Duration").

Fig.17 shows an event handler table (XXX.PROG) that includes an event handler (time events, and user events for menu selection) for each of the playlists.

The event handler table includes the number of defined 5 event handlers/programs ("Number"), and the event handlers/programs (Program#1~Program#n). The content of each of the event handlers/programs includes a definition of an event handler start (<event\_handler> tag) and an event handler ID that matches the ID of the above event, and 10 description in the program is also included between the brackets (i.e. {}) that follow "function".

Next, an internal structure of information (BD.INFO) relating to the entire BD disc will be described with reference to Fig.18.

15 BD disc information is structured from a title list ("TitleList") and an event list ("EventList") relating to global events.

The title list (TitleList) is structured from the number of titles on the disc ("Number"), which is followed 20 by the individual pieces of title information (Title#1~Title#n). Each piece of title information (Title#) is structured from a playlist table ("PLTable") of playlists included in the title, and a chapter list ("ChapterList") of chapters in the title. The playlist table (PLTable) includes

the number of playlists in the title (“Number”), and playlist names (“Name”; i.e. the filenames of the playlists).

The chapter list (ChapterList) is structured from the number of chapters included in the title (“Number”), and individual pieces of chapter information (Chapter#1~Chapter#n). Chapter information (Chapter#) has a table of the cells included in the chapter (“CellTable”), and the cell table (CellTable) is structured from the number of cells (“Number”), and entry information relating to each of the cells (CellEntry#1~CellEntry#k). The cell entry information (CellEntry#) is described by the name of the playlist that includes the cell, and the cell number in the playlist.

The event list (EventList) includes the number of global events (“Number”) and individual pieces of global event information. Here, it should be noted that the first global event defined is referred to as a first event (“FirstEvent”), and this is the event initially called when the BD disc is inserted into a player. The event information for a global event includes only an event type (“Type”) and an event ID (“ID”).

Fig.19 shows a program table (BD.PROG) of a global event handler.

The content of this table is the same as the event

handler table shown in Fig.17.

#### *Event Generation Mechanism*

A mechanism for generating events will now be described  
5 with reference to Figs.20 to 22.

Fig.20 shows an exemplary time event.

As described above, a time event is defined in an event list (EventList) included in playlist information (XXX.PL).  
10 In the case of an event defined as a time event (i.e. Type = TimeEvent), a time event having an ID *Ex1* is sent from a scenario processor to a program processor when an event generation time (*t1*) is reached. The program processor searches for an event handler having an event ID *Ex1*, and executes the target event handler. In the present embodiment, 15 for example, this results in the drawing of two button images.

Fig.21 shows an exemplary user event for conducting a menu operation.

As described above, a user event for conducting a menu operation is also defined in an event list (EventList)  
20 included in playlist information (XXX.PL). In the case of an event defined as a user event (i.e. Type = UserEvent), the user event is rendered into a ready state when an event generation time (*t1*) is reached. At this point in time, the event itself has not yet been generated. The event is in a

ready state for the duration of the time period shown in the valid period information (Duration).

As shown in Fig.21, when a user operates an arrow key (up/down/left/right), RETURN key or the like of a remote controller, firstly a UOP event is generated by a UOP manager, and the generated UOP event is sent to a program processor. The program processor sends the event on to a scenario processor, and on receiving the generated UOP event, the scenario processor conducts a search for a valid event handler, and if a valid event handler exists, the scenario processor generates a user event, and sends the generated user event to the program processor. The program processor searches for an event handler having an event ID *Evl*, and executes the target event handler. In the present embodiment, for example, this results in the playback of playlist#2 being started.

Information relating to which of the remote controller keys was operated by the user is not included in the generated user event. Rather, the program processor obtains information relating to the operated remote controller key from the UOP event, and this information is stored in a SPR register included in a virtual player. It is possible for the program of the event handler to investigate the value of the SPR register, and execute branch processing.

Fig.22 shows an exemplary global event.

As described above, a global event is defined in an event list (EventList) included in information (BD.INFO) relating to the entire BD disc. In the case of an event defined 5 as a global event (i.e. Type = GlobalEvent), an event is only generated when there is a remote controller operation by a user.

When the user operates a MENU key, firstly a UOP event is generated by the UOP manager, and the generated UOP event 10 is sent to a program processor. The program processor sends the generated UOP event on to a scenario processor, which generates a global event and sends the generated global event to the program processor. The program processor conducts a search for an event handler having an event ID *menu*, and 15 executes the target event handler. In the present embodiment, for example, this results in the playback of playlist#3 being started.

In the present embodiment, the remote controller key operated by the user is referred to simply as the MENU key, 20 although there may be, as in DVD, a plurality of MENU keys. In this case, correspondence may be realized by defining IDs that correspond to each of the menus.

### *Virtual Player Machine*

A functional structure of a program processor will now be described with reference to Fig.23.

A program processor is a processing module that 5 includes a virtual player machine. A virtual player machine is a functional model defined as an HD-DVD, and is designed to be applicable with any HD-DVD player. In other words, it is guaranteed that the same functions will be executable, irrespective of the HD-DVD player used.

10 The virtual player machine has two basic operational roles: a programming function and a player variable (register). The programming function is based on Java Script, and the following three operational roles of the programming function are defined as HD-DVD unique functions.

15

1) Link Function: stop current playback, and start playback of designated playlist/cell from designated time

Link: (PL#, Cell#, time)

20 PL# : playlist name

Cell# : cell number

Time : playback start time in cell

2) PNG Draw Function: draw designated PNG data in image

plane  
Draw (File,X,Y)  
File : PNG filename  
X : X coordinate position  
5 Y : Y coordinate position

3) Image Plane Clear Function: clear designated area

of image plane

Clear (X,Y,W,H)

10 X : X coordinate position  
Y : Y coordinate position  
W : X direction width  
H : Y direction width

15 Player variables include system parameters (SPRMs) showing a player status, and general parameters (GPRMs) that are applicable for general use.

Fig.24 lists the system parameters (SPRMs).

SPRM(0) : language code  
20 S P R M ( 1 ) : audio stream number  
SPRM(2) : subtitle stream number  
SPRM(3) : angle number  
SPRM(4) : title number  
SPRM(5) : chapter number

	SPRM(6)	: program number
	SPRM(7)	: cell number
	SPRM(8)	: selected key information
	SPRM(9)	: navigation timer
5	SPRM(10)	: Playback time information
	SPRM(11)	: mixing mode for karaoke
	SPRM(12)	: country information for parental
	SPRM(13)	: parental level
	SPRM(14)	: player configuration (video)
10	SPRM(15)	: player configuration (audio)
	SPRM(16)	: language code for audio stream
	SPRM(17)	: language code for audio stream (extension)
	SPRM(18)	: language code for subtitle stream
15	SPRM(19)	: language code for subtitle stream (extension)
	SPRM(20)	: player region code
	SPRM(21)	: reserved
	SPRM(22)	: reserved
20	SPRM(23)	: playback status
	SPRM(24)	: reserved
	SPRM(25)	: reserved
	SPRM(26)	: reserved
	SPRM(27)	: reserved

SPRM(28) : reserved  
SPRM(29) : reserved  
SPRM(30) : reserved  
SPRM(31) : reserved  
5 S 5  
SPRM(32) : reserved

Here, in the present embodiment, the programming function of the virtual player has a Java Script base, although other programming functions may be used, examples 10 of which include B-Shell (i.e. used in UNIX operating systems, etc) and Perl Script. In other words, the present invention is not restricted to the use of Java Script.

#### *Exemplary Program*

15 An exemplary program of an event handler will now be described with reference to Figs. 25 and 26.

Fig. 25 shows an exemplary menu having two selection buttons.

Using a time event, the program shown in the left half 20 of Fig. 25 is executed at the head of a cell (PlayList#1.Cell#1). Here, the general parameter GPRM(0) is set to "1". GPRM(0) is used in the program to identify a button in a selected state. As a first state, GPRM(0) is set to have an initial value showing button 1 positioned on the left to

be in a selected state.

Next, the function "Draw" is used to draw PNGs relating to buttons 1 and 2. In the case of button 1, the PNG image "1black.png" is drawn with the coordinates (10,200) as the 5 origin (left side). In the case of button 2, the PNG image "2white.png" is drawn with the coordinates (330,200) as the origin (left side).

Furthermore, using a time event, the program shown in the right half of Fig.25 is executed at the end of the cell. 10 Here, a link function is used to designate the playback from the head of the cell again.

Fig.26 shows an exemplary event handler for a user event relating to a menu selection.

When a left/right arrow key or RETURN key of the remote 15 controller is operated, the program corresponding to the operated key is written into the event handler. As shown in Fig.21, when a remote controller key is operated by a user, a user event is generated, and the event handler shown in Fig.26 is started. Branch processing is conducted in the 20 started event handler using the values of general parameter GPRM(0) and system parameter SPRM(8) identifying selected remote controller keys.

Condition 1) "right" key operated with button 1 in

selected state

GPRM(0) is reconfigured to "2", and the selected-state button is changed to button 2 on the right images of buttons 1 and 2 are switched.

5

Condition 2) RETURN ("OK") key operated with button 1 in selected state start playback of playlist#2

10 Condition 3) RETURN ("OK") key operated with button 2 in selected-state start playback of playlist#3

Processing is executed as described above.

15

#### *Player Processing Flow*

Next, the processing flow of the player will be described with reference to Figs.27 to 30.

Fig.27 shows a basic processing flow prior to AV 20 playback.

When the BD disc is inserted (S101), the HD-DVD player reads and analyzes BD.INFO (S102), and reads BD.PROG (S103). BD.INFO and BD.PROG are both temporarily stored in management information storage memory, and analyzed by the scenario

processor.

Next, the scenario processor generates a first event in accordance with first event information (FirstEvent) in BD.INFO (S104). The generated first event is sent to the 5 program processor, and the program processor executes an event handler corresponding to the event (S105).

Information relating to a playlist to be played back first is expected to be stored in the event handler corresponding to the first event. If the playback of a 10 playlist is not instructed, the player waits, without playing back anything, to receive a user event (S201). When the HD-DVD player receives a remote controller operation from a user, the UOP manager has the program manager generate a UOP event 15 (S202).

15 Next, the program manager judges whether the UOP event resulted from the operation of a MENU key (S203), and if "yes", the program manager sends the UOP event to the scenario processor, and the scenario processor generates a user event 20 (S204). The program processor then executes an event handler corresponding to the generated user event (S205).

Fig.28 shows the processing flow from a start of PL playback to a start of VOB playback.

As described above, playlist playback is started by a first event handler or a global event handler (S301). The

scenario processor reads and analyzes playlist information (XXX.PL) as information required to playback the targeted playlist (S302), and reads program information (XXX.PROG) corresponding to the targeted playlist (S303). Next, the 5 scenario processor starts playing back a cell based on cell information stored in the playlist (S304). Cell playback results from the issuing of a request from the scenario processor to the presentation controller, and on receipt of the request, the presentation controller starts AV playback 10 (S305).

When AV playback has been started (S401), the presentation controller reads and analyzes the information file (YYY.VOBI) of the VOB corresponding to the cell to be played back (S402). The presentation controller uses a time 15 map to specify both the VOB to be played back and the address of the targeted VOB, instructs the drive controller to read the specified address, and on receipt of the instruction the drive controller reads the targeted VOB data (S403). The read VOB data is sent to the decoder, and playback is started 20 (S404).

VOB playback is continued until the end of the playback interval of the VOB (S405), and when the end of the VOB playback interval is reached, playback of the next cell is started (S304). If there is no next cell, playback is

terminated (S406).

Fig. 29 shows the flow of event processing from the start of AV playback.

The HD-DVD player is event driven. When playback of a 5 playlist is started, event processing relating to a time event, a user event, and subtitle display is started, and these various event processing operations are executed in parallel.

The S500-series processing steps structure the 10 processing flow for a time event.

After starting playback of a playlist (S501), and judging that the playlist playback has not ended (S502 = "no"), the scenario processor judges whether the time event generation time has been reached (S503). If "yes", the 15 scenario processor generates a time event (S504), and the program processor receives the generated time event and processes a corresponding event handler (S505).

If the time event generation time has not been reached (S503 = "no"), or if the execution of an event handler in step 20 S505 has been completed, the processing returns to step S502 to be repeated. Furthermore, if judged in step S502 that the playlist playback has ended, the time event processing is terminated.

The S600-series processing steps structure the

processing flow for a user event.

After starting playback of a playlist (S601), and judging that the playlist playback has not ended (S602 = "no"), it is judged whether a UOP has been received (S603). If "yes", 5 the UOP manager generates a UOP event (S604), and on receipt of the generated UOP event, the program processor judges whether the UOP event is a menu call (S605), and if "yes", the program processor has the scenario processor generate an event (S607), and the program processor executes a 10 corresponding event handler (S608).

If judged in step S605 that the UOP event is not a menu call, this shows that the UOP event is an event resulting from the operation of an arrow key or a RETURN key. In this case, the scenario processor judges whether the present time 15 is within the valid time period (S606), and if "yes", the scenario processor generates a user event (S607), and the program processor executes a corresponding event handler (S608).

If a UOP has not been received (S603 = "no"), or if the 20 present time is not with the valid period (S606 = "no"), or if execution of an event handler in step S608 has ended, then the processing returns to step S602 to be repeated. Furthermore, if judged in step S602 that the playlist playback has ended, the user event processing is terminated.

Fig.30 shows a subtitle processing flow.

After starting playback of a playlist (S701), and judging that the playlist playback has not ended (S702 = "no"), it is judged whether a subtitle draw time has been reached 5 (S703). If "yes", the scenario processor instructs the presentation controller to draw a subtitle, and the presentation controller has the image processor draw a subtitle (S704). If judged in step S703 that the subtitle draw time has not been reached (S703 = "no"), it is judged 10 whether a subtitle display end time has been reached (S705). If "yes", the presentation controller sends a subtitle-erase instruction to the image processor, and the drawn subtitle is erased from the image plane (S706).

If the subtitle draw step S704 is completed, or if 15 subtitle erase step 706 is completed, or if judged in step S705 that the subtitle display end time has not been reached, the processing returns to step S702 to be repeated. Furthermore, if judged in step S702 that the playlist playback has ended, the subtitle display processing is 20 terminated.

#### Embodiment 2

An embodiment 2 of the present invention will now be described.

Embodiment 2 relates to a BD player having a plurality of execution-related faculties. Embodiment 2 is based on embodiment 1, and the following description focuses on extensions to and differences with embodiment 1.

5

#### *The Concept of a Virtual Package*

Fig.31 is a conceptual diagram of a virtual package according to the present invention.

Conventionally, an "application format" was a format dependant on physical media such as CD, DVD, or BD (HD-DVD) as described in embodiment 1. A virtual package as described in the present invention is a new application based on a new format not dependant on physical media.

As shown in Fig.31, a virtual package according to the invention is presented to a user as a single package comprising the dynamic combination of a variety of contents, such as contents on a BD disc, contents on a recording medium such as a hard disk, a memory card or the like, and contents on the Internet. Contents on a hard disk, a memory card or the like may, for example, be contents downloaded from the Internet and stored on a local storage device (i.e. hard disk, memory card, etc), or contents copied from a friend.

Fig.32 shows an exemplary application using a virtual package.

A Hollywood movie is illustrated as an example. In the movie industry there exists a business model known as "timeshift". According to this business model, a movie is first released in the United States, before being released 5 successively by region, starting, for example, in Western Europe, and then moving on to Japan, Eastern Europe, and Asia. This business model is premised on the need for translation and other processes that result from differences in languages between the various regions/countries.

10 Movies produced in the United States will almost certainly be in English, and thus for the movie to be screened in Japan, for example, either subtitles or dubbing must be provided. Generally the translation into Japanese is conducted after the movie has been made, and a Hollywood movie 15 is naturally released in the United States prior to being released in Japan.

As in the movie industry, the preparation of Japanese material in the package industry is also generally conducted after the English title has been released, and in this case 20 there is likely to be a delay between the release of the title in the United States and the release of the Japanese version.

However, the Japanese title is generally only made available in Japan, and thus viewing the Japanese dubbing/subtitles on a package purchased in the United States

will, of course, not be possible because the data for the Japanese title does not exist on the original disc.

With a virtual package, however, it is possible to download data onto a hard disk or a memory card via the 5 Internet after the Japanese version has been completed, and by integrating this data when a BD disc is played back, a user can enjoy the Japanese version of the title by using the English version on the BD disc in conjunction with the virtual package containing the Japanese language material.

10 As shown in Fig.32, a virtual package does not only accommodate differences in language, but also allows for the downloading of supplementary data such as games, trailers, director/actor commentary, the making of, and video clips.

15 A "language credit" for switching the language of a playback scene (video) will now be illustrated as an exemplary virtual package.

At the top of Fig.33 is shown a conventional package stored on a BD disc, and below that is shown a virtual package. The virtual package includes a PlayList#4, and based on a 20 language setting by the user, an automatic selection is made as to whether to play PlayList#2, which includes the scene in English, or PlayList#4, which includes the Japanese version of the scene.

Although discussed in detail in a later section, in

order to realize this structure, it is necessary to provide the following files additionally to those on the BD disc: 004.PL/004.PROG for PlayList#4; VOB files referred to by PlayList#4, and VOBI files for the VOB files; 001.PROG 5 (program file) for branch processing executed in PlayList#1.

As shown in Fig.31, these files can be stored on storage media such as a memory card, a hard disk drive, or the like, or downloaded directly for use from the Internet.

Fig.34 shows an exemplary parental lock. In the given 10 example the original package on the BD disc is a linear movie title from PlayList#1 to PlayList#3. However, because PlayList#2 includes a sexual/violent scene, the title contains a viewing age restriction of the type typified by "PG18" (parental guidance recommended) and the like.

15 By downloading a child-oriented scene (i.e. material replacing the violet/sexual scene) as a PlayList#5, storing the downloaded scene on a memory card, hard disk or similar storage media, and playing a virtual package that combines the contents of the BD disc and the contents on the storage 20 media, the title can be altered so as to be enjoyable to both adults and children.

Furthermore, in Fig.34, a PlayList#4 is also downloaded as a menu for selecting whether a particular scene is for adults or for children.

Fig.35 shows an exemplary director's cut.

Sometimes scenes from a movie are cut or edited so as to reduce the running time of the movie to a two-hour time period, for example. This is done to increase the number of 5 screenings per day, and is often contrary to the wishes of the director. Now in the age of technology such as DVD, it is increasingly common to come across storage media containing unreleased scenes cut from a cinema-release version of a movie.

10       Needless to say, unreleased scenes, scenes cut in the editing process, and the like, are of value to movie buffs, who would be extremely happy to obtain such material. Here, the present invention allows for unreleased scenes and the like, downloaded either directly or indirectly onto storage 15 media such as a hard disk, a memory card or the like via the Internet, to be played back as a virtual package in conjunction with original material on a BD disc.

In Fig.35, PlayList#3 is an unreleased scene, and is not included in the original version on the BD disc.

20       In the above example, a new playlist and VOB is added to provide new video and audio data. However, rather than adding a new playlist, it is possible, as shown respectively in Figs.36 and 37, to add only material or to replace a playlist.

Fig.36 shows an exemplary application of the downloading of supplementary material such as commentary and Japanese language audio data. Here, only audio data is downloaded rather than a playlist.

5 Fig.37 shows an exemplary addition of a new angle to a multi-angle interval. Here, a playlist is replaced rather than being added. Furthermore, VOBs and VOB information required in the angle playback are also supplemented to form a virtual package.

10

#### *Generation of a Virtual Package*

Fig.38 is a structural diagram showing directories and files on a hard disk or a memory card.

15 Below a ROOT directory, is placed a "VPACKAGES" directory for storing virtual packages, and in this directory are placed "ABC", "XYZ" and other directories relating to particular titles. These directory names match the package names of packages on the BD disc. Furthermore, below the directories of each package are placed "VERSION2",  
20 "VERSION3" and other directories relating to each package.

Below each package directory is stored, as a virtual package, the contents supplementing the BD original package. The following files are stored in the given example:

	BD.INFO	virtual package management information
	BD.PROG	virtual package global event handler
	002.PL	PlayList#2 management information
	002.PROG	PlayList#2 event handler
5	003.PL	PlayList#3 management information
	003.PROG	PlayList#3 event handler
	002.VOBI	VOB#2 management information
	002.VOB	VOB#2 stream

10 Next, the processing prior to and including the generation of a virtual package will be described with reference to Fig.39. Fig.39 shows the generation of a virtual package using the downloaded contents in Fig.38.

15 As shown in Fig.39, the following six files are stored on the original disc.

	BD.INFO	BD package management information
	BD.PROG	BD package global event handler
	001.PL	PlayList#1 management information
20	001.PROG	PlayList#1 event handler
	001.VOBI	VOB#1 management information
	001.VOB	VOB#1 stream

The BD disc is structured from a single playlist and

a single VOB, the contents of which are the same as described in embodiment 1.

The VERSION2 directory shown in Fig.38 contains the following four files:

5

BD.INFO	VERSION2 management information
BD.PROG	VERSION2 global event handler
002.PL	PlayList#2 management information
002.PROG	PlayList#2 event handler

10

A new playlist as well as management information and a global event handler for the virtual package is stored, and as a result, the virtual package is constituted as follows:

15

BD.INFO	VERSION2 management information (from VERSION2 directory)
BD.PROG	VERSION2 global event handler (from VERSION2 directory)
20	001.PL PlayList#1 management information (from BD disc)
	001.PROG PlayList#1 event handler (from BD disc)
	002.PL PlayList#2 management information (from VERSION2 directory)

002.PROG PlayList#2 event handler (from VERSION2  
directory)  
001.VOBI VOB#1 management information (from BD  
disc)  
5 001.VOB VOB#1 stream (from BD disc)

A VERSION3 of the virtual package will now be described.

10 The VERSION3 directory stores eight files (see Fig. 38),  
and as a result, VERSION3 of the virtual package is  
constituted as follows:

15 BD.INFO VERSION3 management information (from  
VERSION3 directory)  
BD.PROG VERSION3 global event handler (from  
VERSION3 directory)  
001.PL PlayList#1 management information (from BD  
disc)  
001.PROG PlayList#1 event handler (from BD disc)  
20 002.PL PlayList#2 management information (from  
VERSION3 directory)  
002.PROG PlayList#2 event handler (from VERSION3  
directory)  
003.PL PlayList#3 management information (from

VERSION3 directory)  
003.PROG PlayList#3 event handler (from VERSION3  
directory)  
001.VOBI VOB#1 management information (from BD  
disc)  
5 001.VOB VOB#1 stream (from BD disc)  
002.VOBI VOB#1 management information (from  
VERSION3 directory)  
002.VOB VOB#2 stream (from VERSION3 directory)

10

As shown above, new content is added to the original content on the BD disc in order to constitute a virtual package based on different versions.

As with Fig.38, Fig.40 shows a structure of directories  
15 and files stored on a hard disk or a memory card, although in this case, each package does not have a separate directory. The files to be overwritten are, as shown in Fig.39, determined at the time a virtual package is dynamically generated, and thus it is possible to validate the latest  
20 version of files having the same filename. Here, it is of course possible to also overwrite files when data for a virtual package is downloaded.

*Data Structure of a Virtual Package*

Fig.41 is a structural diagram of a BD.INFO file containing management information relating to an entire virtual package.

5        Except for the addition of a few fields, this management information is basically the same as the management information for the entire BD disc described in embodiment 1. Furthermore, the management information stored on the BD disc in embodiment 2 relates to a virtual package rather than 10 a BD disc as in embodiment 1.

Management information (BD.INFO) relating to a virtual package includes four fields: general information ("General"), a PL list ("PLList"), a title list ("TitleList"), and an event list ("EventList").

15        Of these four fields, the general information (General) includes a package name ("PackageName"), a packet version ("PackageVersion"), a downloadability flag ("Downloadability"), a URI ("URI"), and an expiration date ("ExpirationDate").

20        Here, the package name (PackageName) shows the name shared by the BD package and the virtual package (the movie title is generally used). The package name is later used to establish the correspondence between the BD package and the virtual package.

The packet version (PackageVersion), in the case of the BD package, shows "VERSION1", and in the case of the virtual package, shows "VERSION2", "VERSION3", ..., as the case may be. In other words, The packet version identifies the version 5 of the virtual package to which the BD.INFO file belongs.

The downloadability flag (Downloadability) shows whether or not there are contents to be downloaded. In the case of a BD package, if the downloadability flag is "off", this shows that a virtual package does not exist. On the other 10 hand, in the case of a virtual package, if the downloadability flag is "off", this shows that additional versions to the current version cannot be generated.

The URI (Unified Resource Identifier) shows, in the case of a BD package, the location of a virtual package. The 15 location shown by the URI can thus be accessed, and the latest version of the virtual package obtained.

The expiration date (ExpirationDate) shows the date up until which it is possible to download the virtual package. If the player has date-time information, the expiration date 20 can be compared with the present date-time, and depending on the current validity status of the virtual package, a judgment can be made as to whether to try and obtain contents from a server shown in the URI.

The PL list (PLList) field in BD.INFO shows the number

of playlists ("Number") included in the package, as well as playlist entries (PLEntry#) for each of the playlists.

Here, each playlist entry (PLEntry#) shows a package version ("PackageVersion"), a storage name ("StorageName"), 5 a playlist name ("PlayListName"), and alternative playlist information ("Alternative").

Here, the package version (PackageVersion) shows the version in which a particular playlist is stored. For example, if the playlist on the BD disc is used in its original state, 10 the package version shows "VERSION1". However if the playlist is stored in VERSION3 of the virtual package, the package version shows "VERSION3".

The storage name (StorageName) shows where a particular playlist is located. For example, the storage name may 15 distinguish between a BD disc, a hard disk, a memory card, or an Internet server. Since the identification of a hard disk or a memory card is conducted in the player, the storage name is automatically provided by the system after the downloading of a playlist as a virtual package.

20 The playlist name (PlayListName) shows the name of the playlist (e.g. Playlist#1, etc).

The alternative playlist information ("Alternative") shows an alternative playlist to be played if, for example, the target playlist is unavailable due to a downloading

failure, a hard disk error, or the like.

Fig.42 shows a playlist file structure according to embodiment 2.

The playlist file (XXX.PL) in embodiment 2 is an extension of the playlist file described in embodiment 1, and includes data used in both the BD package and the virtual package. The following description focuses on the differences with embodiment 1.

Cell information (Cell#) in a playlist file according to the present embodiment can refer to a main VOB containing video data, and VOBs containing supplementary audio data. Here, the first VOB entry (VOBEntry#1) stores the main video VOB, whereas subsequent VOB entries are only permitted to store audio VOBs. The total number of VOBs referred to by a cell is obtained by summing together the main VOB and the respective number of audio VOBs. This information ("Number") is stored at the head of a cell, after which is stored the VOB entries (VOBEntry#), an audio table ("AudioTable"), and a subtitle table ("SubtitleTable").

Each VOB entry includes a VOB name (VOBName#; e.g. "001.VOB"), a VOB start time ("In"), a VOB end time ("Out"), and offset information ("Offset").

Here, the offset information (Offset) is information used to adjust the system clocks between the main VOB and

audio-only VOBs. In order to realize AV synchronization, MPEG streams include a timestamp. Although a master clock for generating the timestamp is wholly contained (i.e. continuous) within a single MPEG stream, it is not normally possible to achieve coordination between the timestamps in two different MPEG streams because of the two streams containing separate master clocks. Here, synchronization between the two MPEG streams is realized by including the difference between the master clocks as offset information.

The audio table (AudioTable) in a cell includes, from a head of the table, information showing the number of audio streams handled by the playlist ("Number"), and the individual audio entries (AudioEntry#). Each audio entry has a VOB number ("VOBNumber") of the VOB containing the audio data, and a stream number ("StreamID") of the stream in the VOB.

The subtitle table (SubtitleTable) in a cell includes language information (Language#), which in turn contains information (i.e. "StorageName" as in embodiment 1) showing the storage location of a subtitle. The storage name (StorageName) contains information relating to a device (e.g. BD disc, hard disk, Internet, etc) to be accessed in order to obtain the subtitle.

Fig. 43 shows a structure of VOB management information

according to embodiment 2.

The VOB management information in embodiment 2 is an extension of the VOB management information described in embodiment 1, and is data used for both the BD package and 5 the virtual package. The following description focuses on the differences with embodiment 1.

The VOB management information is characterized firstly, as described above, by the existence of VOBs that do not contain video data. In the VOB management information 10 there is an area for a video data attribute to be shown, although in the case of VOBs not containing video data, the "Video" field in the attribute information ("Attribute") is set to shows "no video" and then not referred to again.

In the present embodiment, offset information 15 ("Offset") is also contained in the audio attribute information (Audio#).

Here again, the offset information (Offset) is information used to adjust the system clocks between the main VOB and the audio-only VOBs. As described above, in order 20 to realize AV synchronization, MPEG streams include a timestamp. Although a master clock for generating the timestamp is wholly contained (i.e. continuous) within a single MPEG stream, it is not normally possible to achieve coordination between the timestamps in two different MPEG

streams because of the two streams having separate master clocks. Here, synchronization between the two MPEG streams is realized by including the difference between the master clocks as offset information.

5 A detailed description of the offset information will now be given with reference to Fig.44.

001.VOB in the lower half of Fig.44 contains the main VOB having video data, and 101.VOB in the upper half of Fig.44 contains an additional VOB having only audio data. A VOB is 10 constituted by an MPEG program stream, and a stream is constituted as a series of packs. Each pack contains an SCR timestamp as synch information. Here, the offset information shows the difference between the SCRs in different packs.

15 For example, if the offset value is 252, then the SCR value of an audio pack in 101.VOB that corresponds to an audio pack in 001.VOB having a SCR of 352, will be 100. The correspondence between audio packs is information used in the sync playback conducted by the player.

20 In the above description, SCR timestamps are used as a base, although an SCR is timing information inputted to a pack by a multiplexer, and thus it is not always the case that two VOBs will contain packs sharing the same value. Instead of SCRs, PTS timestamps may be used as a base. A PTS (presentation timestamp) is playback time information

included in an audio pack. As with SCRs, it is possible to use PTSS to determine the correspondence between audio data.

A time map in an audio-only VOB will now be described with reference to Fig.45.

5 As described in embodiment 1, a VOB is managed in units (VOBUS) based on video GOPs. However, since audio-only VOBS do not have equivalent units, it is not generally possible to structure VOBUS.

10 In the present embodiment, however, VOBUS in audio-only VOBS are structured on the basis of corresponding video VOBS. In other words, the VOBUS in an audio-only VOB are made to correspond to audio data included in the VOBUS of a video VOB.

15 Thus, using a video VOB as a basis, the VOBUS of an audio-only VOB are structured so as to include only audio data that is the same length as audio data in the VOBUS of the video VOB.

#### *Player Structure*

20 Fig.46 is a schematic diagram of a player for playing a virtual package according to the present embodiment.

In addition to the elements shown in Fig.6, the player in embodiment 2 includes a hard disk 2001, a reading control unit 2002, a virtual package storage unit 2003, and a virtual

package generation unit 2004.

Hard disk 2001 stores an additional file for structuring a virtual package. Reading control unit 2002 controls the reading of virtual package data from either hard disk 2001 or BD disc 201. Virtual package generation unit 5 2004 generates a virtual package based in the virtual package generation method described above, and stores the generated virtual package in virtual package storage unit 2003.

Fig.47 is a functional block diagram showing AV data 10 processing in a player for playing a virtual package.

The player in Fig.47 is structured from a hard disk 4001, a BD disc 4002, a track buffer 4003, an audio buffer 4004, a first image memory 4005, a second image memory 4006, a demultiplexer 4007, an image processor 4008, a video 15 processor 4009, an audio processor 4010, a clock 4011, and an adder 4012.

AV data in BD disc 4002 is temporarily stored in track buffer 4003, and then sent to first image memory 4005 in the case of subtitle/graphics data, and to demultiplexer 4007 20 in the case of main video data or audio data (i.e. MPEG stream). On the other hand, AV data on hard disk 4001 is sent to second image memory 4006 in the case of subtitle/graphics data, and to demultiplexer 4007 via audio buffer 4004 in the case of audio data.

The subtitle/graphics data stored in the first and second image memories is sent to image processor 4008, where it is decoded.

5 The MPEG stream sent from the track and audio buffers is demultiplexed by demultiplexer 4007. The demultiplexed video and audio data is then sent to video processor 4009 and audio processor 4010, respectively, where it is decoded and outputted.

10 The image and video data decoded by the image and video processors, respectively, is integrated and outputted by adder 4012.

Clock 4011 generates a reference clock to be used in the sync playback of AV data.

15 Fig.48 shows a detailed structure of the demultiplexer and the audio processor in Fig.47.

The upper and lower halves of Fig.48 show different structures of both demultiplexer 4007 and audio processor 4010.

20 Demultiplexer 4007, as shown in the upper half of Fig.48, is structured from a first buffer 4101 for temporarily storing data inputted from track buffer 4003, a second buffer 4102 for temporarily storing data inputted from audio buffer 4004, a filter 4103 for filtering data inputted from the first and second buffers, and a demultiplexer core 4104.

Demultiplexer 4007, as shown in the lower half of Fig. 48, is structured from a first demultiplexer core 4202 for demultiplexing data inputted from track buffer 4003, a second demultiplexer core 4203 for demultiplexing data inputted from audio buffer 4004, and an offset 4201 for realizing sync playback between the two demultiplexer cores. Here, offset 4201 receives input of an offset value in the data structure.

5 Audio processor 4010, as shown in the lower half of Fig. 48, is structured from a first audio processor core 4204 for decoding audio data sent from first demultiplexer core 4202, a second audio processor core 4205 for decoding audio data sent from second demultiplexer core 4203, and an adder/switcher 4206 for integrating or switching the audio data decoded by the two audio processor cores.

15

#### *Player Processing Flow*

The processing flow performed by a player will now be described starting with Fig. 49.

When a BD disc is inserted into a player for playing 20 a virtual package (S8001), the player instructs reading control unit 2002 to read BD.INFO from the BD disc (S8002). Next, the player judges via the Internet whether the BD.INFO specified by the URI exists on a server (S8003). If "no", the processing moves to step S8007.

On the other hand, if judged in step S8003 that the specified BD.INFO does exist on a server (S8003 = "yes"), the player obtains the specified BD.INFO from the server (S8004), and compares the content of the two BD.INFO files 5 in order to judge whether there is new content. If there is new content, the player downloads the new content, stores the downloaded content on a hard disk or a memory card (S8006), and starts playback processing of the virtual package (S8008). On the other hand, if judged that there is no new content 10 (S8005 = "no"), the processing moves to step S8007.

In step S8007, the player judges whether virtual package BD.INFO corresponding to the BD disc BD.INFO exists on the hard disk. If "yes" the player starts playback of the virtual package (S8008), and if "no" the player starts normal 15 playback of the BD disc (S8009). Here, normal playback of the BD disc is the same as described in embodiment 1.

As described in relation to Fig.39, the virtual package played in step S8008 is generated by virtual package generation unit 2004, and stored in virtual package storage 20 unit 2003.

Fig.50 is a flowchart of basic virtual package processing.

When playback of a virtual package is started (S1001), the player uses reading control unit 2002 to read BD.INFO

from the hard disk, and sends the BD.INFO to virtual package generation unit 2004 (S1002). Next, the player again uses reading control unit 2002 to read BD.PROG from the hard disk, and sends the BD.PROG to virtual package generation unit 2004 (S1003).

The processing that follows step S1003 is the same as that described in relation to Fig.27, and a description is omitted here.

Fig.51 is a flowchart of the playback processing of a playlist in the virtual package.

When playback of the playlist is started (S3001), the player firstly judges whether the playlist actually exists on either the hard disk or the BD disc (S3002). If "no", the playlist is switched for an alternative playlist (S3003).

Next, the player judges whether the target playlist (XXX.PL) is on the hard disk or on the BD disc (S3004). If the playlist data is on the hard disk, the player uses reading control unit 2002 to read XXX.PL (S3005) and XXX.PROG (S3006) from hard disk 2001.

On the other hand, if the playlist data is on the BD disc, the player uses reading control unit 2002 to read XXX.PL (S3007) and XXX.PROG (S3008) from the BD disc.

After reading the respective files, the player starts playing a cell (S3009), and then starts playing AV data

corresponding to the cell (S3010).

Fig.52 is a flowchart of the playback processing of AV data in a virtual package.

After starting the AV playback (S4001), the player 5 judges whether the target VOB (YYY.VOB) is on the hard disk or on the BD disc (S4002). If the VOB data is on the hard disk, reading control unit 2002 starts reading YYY.VOBI (S4003) and YYY.VOB (S4004) from hard disk 2001.

On the other hand, if the VOB data is on the BD disc, 10 reading control unit 2002 reads YYY.VOBI (S4005) and YYY.VOB (S4006) from the BD disc.

Next, the player judges whether the playlist currently being played contains audio data to be played in synchronization with the read VOB (S4007). If "yes", reading 15 control unit 2002 starts reading, from the hard disk, VOB information ZZZ.VOBI relating to the audio data (S4008) and the audio VOB data ZZZ.VOB (S4009), and the player starts sync-playback processing of the VOBs (S4010).

On the other hand, if there is no audio data to be sync 20 played (S4007 = "no"), the player starts normal playback of the VOB (S4011). Here, normal playback of a VOB is the same as described in embodiment 1.

Fig.53 is a flowchart of the sync-playback processing of VOBs in a virtual package.

When VOB sync-playback processing is started (S9001), the player starts reading YYY.VOB (i.e. VOB file containing main video data) into the track buffer (S9002), and at the same time starts reading ZZZ.VOB (i.e. audio-only VOB file) into the audio buffer (S9003). The player waits until the track buffer is full (S9004), and when full, the player starts playback of YYY.VOB (S9005). Next, the player refers to the clock difference shown in the offset, and waits for a sync timing of the audio data that is to be sync-played (S9006). 5 When the sync timing is reached, the player starts audio playback of the audio data ZZZ.VOB. 10

Once playback of the respective VOBs has been started, steps S9008 to S9010 are repeated until the completion of the VOB playback.

15 In step S9008, the player uses the value of system parameter SPRM(1) to judge whether the audio data currently being played is YYY.VOB or ZZZ.VOB audio data. If the audio data currently being played is YYY.VOB audio data (i.e. audio data multiplexed with video data on the same VOB), the 20 processing moves to step S9011. On the other hand, if the audio data currently being played is judged to be ZZZ.VOB audio data, in step S9009 the player judges, as shown in Fig.44, whether YYY.VOB and ZZZ.VOB contain audio pack having matching SCRs (by referring to the offset value). If matching

SCRs exist, the player uses filter 4103 to exchange the audio packs having matching SCRs (9010).

When judged that VOB playback has ended (S9011 = "yes"), the player judges whether there is a next cell to be played 5 (S9012), and if "yes" the player starts playing the next cell (S9013).

Another flowchart of sync-playback processing of VOBs in a virtual package will now be described with reference to Fig.54.

10 The flowchart in Fig.53 shows processing that corresponds to the demultiplexer described in the upper half of Fig.48, whereas the flowchart in Fig.54 shows processing that corresponds to the demultiplexer described in the lower half of Fig.48.

15 The processing in steps S9101 to S9108 in Fig.54 is the same as that in steps S9001 to S9008 in Fig.53. The second flowchart of sync-playback processing as shown in Fig.54 differs from step S9109 onwards.

If judged in step S9108 that the audio data to be played 20 is in YYY.VOB, switcher 4206 in Fig.48 outputs the YYY.VOB audio data decoded by first audio processor 4204. On the other hand, if judged that the audio data to be played back is in ZZZ.VOB, switcher 4206 outputs the ZZZ.VOB audio data decoded by second audio processor 4205.

Step S9111 is for judging whether the VOB playback has ended, and corresponds to step S9011 in Fig.53. Steps S9112 and below in Fig.54 are the same as steps S9012 and below in Fig.53, and will not be described here.

5 Fig.55 is a flowchart of event processing during the playback of a playlist in a virtual package.

Steps S6001 to S6008 are the same as steps S601 to S608 in the Fig.29 flowchart, and will not be described here.

10 The Fig.55 flowchart differs in that if judged in step S6006 that the user event is not valid (i.e. request received from user is not a user event), the processing proceeds to steps S6009 and below, rather than returning to step S6002.

15 In step S6009, the player judges whether the request from the user is to switch audio data. If "yes", in step S6010 the player sums SPRM(1) (i.e. system parameter managing the audio stream number). Next, in step S6011 the player judges whether the audio stream number is valid (i.e. whether corresponding audio data exists). If corresponding audio data exists does not exist, (S6011 = "yes"), SPRM(1) is reset  
20 (S6012).

Next, the player judges whether there is a subtitle-switching request (S6013). If "yes", in step S6014 the player sums SPRM(2) (i.e. system parameter managing the subtitle number), and judges whether a corresponding

subtitle stream exists (S6015). If a corresponding subtitle stream does not exists (S6015 "yes"), SPRM(2) is reset in step S6016.

After completing the above steps, the processing 5 returns to step S6002 and the player judges whether playlist playback has ended.

Fig.56 is a flowchart of processing to switch a subtitle during playback of a playlist in a virtual package.

When playback of the playlist is started (S7001), steps 10 S7002 to S7013 are repeated.

In step S7002, the player judges whether the playlist playback has ended, and repeats the subsequent processing until the completion of the playlist playback.

In step S7003, the player judges whether in the image 15 memory there is a PNG whose draw interval has ended, and if "yes" the PNG data in the image memory is discarded (S7004). Next, the player judges whether there is a PNG whose subtitle start time has been reached (S7005), and if "yes" the player draws the subtitle (S7006), and returns to step S7002.

If in step S7005 there is no PNG whose subtitle start 20 time has been reached (S7005 = "no"), the player judges whether there is a subtitle-switching request (S7007), and if "yes" the player judges whether there is a subtitle currently being displayed (S7008). If "yes" the player erases

the subtitle currently being displayed (S7009).

Next, the player judges whether the target subtitle stream contains a subtitle whose display period has been reached (S7010), and if "yes" the player draws the subtitle 5 (S7011).

The player then judges whether the display end time of the subtitle currently being displayed has been reached (S7012), and if "yes" the player erases the subtitle currently being displayed (S7013).

10

### Embodiment 3

An embodiment 3 of the present invention will now be described.

Embodiment 3 relates to the handling of a plurality of 15 virtual packages. Embodiment 3 is based on embodiments 1 and 2, and the following description focuses on extensions to and differences with embodiments 1 and 2.

Fig.57 is an image diagram of playback selection involving a plurality of virtual packages.

20 As shown in Fig.57, the user is able to select a desired package for playback from among a number of virtual packages including the original package stored on the BD disc.

Fig.58 is a schematic block diagram of a player for making a playback selection from a plurality of virtual

packages.

Fig.58 shows an extension of the player in Fig.46 of embodiment 2. Specifically, the player additionally includes a virtual package display control unit 2005 for drawing/displaying virtual package information generated by virtual package generation unit 2004 in image plane 209. Furthermore, virtual package display control unit 2005 receives a user event from program processing unit 206, receives notification of which package the user has selected for playback, and assists in the playback of the selected virtual package in virtual package generation unit 2004.

Fig.59 is a variable table showing information relating to the playback of a virtual package.

This table is an extension of the table shown in Fig.24 of embodiment 1, and additionally includes package information relating to a selected virtual package in system parameter SPRM(24).

Fig.60 is a structural diagram of information for managing a plurality of virtual packages. This information is stored in respective directories (i.e. having the same structure as in embodiment 2) of the virtual packages.

Virtual package management information "VP.INFO" includes a BD table for the BD.INFO ("BDTable"). The BD table contains the number of BD entries ("Number") as well as the

individual BD entries (BDEntry#). Each BD entry stores corresponding BD.INFO information.

Fig.61 is a flowchart of the processing prior to and including the selection of a virtual package from a plurality of virtual packages.

When a BD disc is inserted (S10001), virtual package generation unit 2004 generates virtual package information (S10002). The generated information is displayed on a screen by virtual package display control unit 2005 in order to show selectable virtual packages (S10003), and the player then waits for a user input (S10004). On receipt of a virtual package selection from the user, the player stores the selected virtual package information in virtual package storage unit 2003 (S10005), and starts playback of the corresponding virtual package (S10006).

In the case of a plurality of virtual packages, the virtual package information can, as described above, be generated in the player, or alternatively, this information may be provided by a contents provider.

20

#### Embodiment 4

An embodiment 4 of the present invention will now be described.

Embodiment 4 relates to the handling of a plurality of

virtual packages via a home network or the like. Embodiment 4 is based on embodiments 1, 2 and 3, and the following description focuses on extensions to and differences with embodiments 1, 2 and 3.

5 Fig.62 is a schematic structural diagram of a player for playing a virtual package according to embodiment 4.

The player in embodiment 4 is an extension of the player shown in Fig.46, and additionally includes a virtual package table 2006.

10 Fig.63 is a structural diagram showing virtual package table information managed in a virtual package table 2006.

The virtual package table information is structured from the package names of each of the virtual packages, storage position information relating to each of the virtual 15 packages, and end position information showing the previous packages, and end position information showing the previous playback end position of the virtual packages.

20 In the case of a package "ABC", for example, the storage position is an "ABC" directory on a local hard disk, and an end position of the previous playback was 3 minutes 24 seconds into cell#1 of playlist#2.

In the case of a package "XYZ", there is no storage position information, which means that the package is not a virtual package but an original package stored on the BD disc.

A package "WWW" is stored in "Memory: ¥WWW", which means that the package is in a "WWW" directory on a memory card. Furthermore, there is no end position information, which means that the package is yet to be played.

5 Finally, a package "JKF" is stored in a "JKF" directory on a home server.

Fig.64 is a flowchart of processing prior to a start of a virtual package playback according to the present embodiment.

10 When a BD disc is inserted (S11001), the player judges whether corresponding virtual package information exists in the virtual package table information (S11002).

15 If judged that corresponding virtual package information does not exist (S11002 = "no"), the player conducts normal BD playback (S11012), and when the BD playback ends (S11013), the player generates virtual package information and updates the virtual package table (S11014).

20 If judged that corresponding virtual package information does exist (S11002 = "yes"), the player obtains storage position information from the virtual package table (S11003), and judges whether valid data is stored in the storage position information (S11004).

If judged in step S11004 that the storage position information is invalid (S11004 = "no"), the player conducts

normal BD playback (S11009), and when the BD playback has ends (S11010), the player updates the virtual package information and the virtual package table (S11011).

On the other hand, if judged in step S11004 that the 5 storage position information is valid (S11004 = "yes"), the player obtains BD.INFO from the location shown in the storage position information (S11005), and plays the virtual package (S11006). When the virtual package playback ends (S11007), the player updates the end position information (S11008).

10 Fig.65 is a flowchart of processing at a start of the virtual package playback according to embodiment 4.

When playback of a virtual package is started (S1101), reading in of management information (BD.INFO) and a global event handler (BD.PROG) relating to the package is conducted 15 in steps S1102 and S1103, respectively. The player then starts playback of the virtual package from the position shown in the end position information (S1104). Here, it should be noted that the first event handler processing activated by BD.INFO and BD.PROG is normally invalid.

20 Fig.66 shows exemplary playback in a home network and a stand-alone. Although the description in the present embodiment focuses on playback conducted in a home network, it is not always the case that the player is connected to a home network. As such, the following description relates

to processing steps conducted when a stand-alone is used to conduct playback.

The lower half of Fig.66 shows an exemplary virtual package playback player connected to a home network, whereas 5 the upper half of Fig.66 shows both a virtual package playback player in which a hard disk is internally provided, and a virtual package playback player which is memory card compatible.

Fig.67 is a flowchart of processing prior to a start 10 of virtual package playback in a virtual package playback player according to any of the plurality of structures given above.

When a BD disc is inserted (S12001), the player makes 15 a judgment as to its (i.e. the player's) own structure. If in step S12001 the player judges that it (i.e. the player) has an internal hard disk, and that the hard disk is connected, the player judges whether there is virtual package information on the hard disk (S12003). If "yes" the processing moves to step S12008, and if "no" the processing 20 moves to step S12004.

In step S12004, the player judges whether it (i.e. the player) can connect to a server. If "yes", the player judges whether there is virtual package information on the server (S12005). If "yes" the processing moves to step S12008, and

if "no" the processing moves to step S12006.

In step S12006, the player judges whether there is a memory card in the player. If "yes" and if judged in step S12007 that there is virtual package information in the 5 memory card (S12007 = "yes"), the processing moves to step S12008. However, if there is no memory card (S12006 = "no") or if there is no valid virtual package information on the memory card (S12007 = "no"), the player starts normal BD playback (S12010).

10 In step S12008, the player reads virtual package information, and starts playing the virtual package (S12009).

In the present embodiment, the processing proceeded from a hard disk to a server and then on to a memory card, 15 although it is possible to vary this order. For example, the possibility of connecting to a server may be judged first, before moving on to judge whether there is a memory card in the player, and finally if the player has an internal hard disk.

20

#### Embodiment 5

An embodiment 5 of the present invention will now be described.

Embodiment 5 relates to the handling of a plurality of

virtual packages via a home network or the like. Embodiment 5 is based on embodiments 1 to 4, and the following description focuses on extensions to and differences with embodiments 1 to 4.

5 Fig.68 shows a structure of a multi-angle. The upper half shows an original package, and the lower half shows a virtual package.

10 In the upper half of Fig.68, a playlist#2 is an angle interval structured from two VOBs (VOB2, VOB3). The two VOBs are interleaved stored in the angle interval. Here, interleaved storage is substantially the same as the storage of image data described in relation to Figs.11 and 12 in embodiment 1.

15 The virtual package in the present embodiment involves the downloading of a VOB5 as a new third angle, with respect to the BD disc. The virtual package is constituted by new playlist information for playlist#2, as well as VOB data and VOB information for VOB5, all of which is downloaded and stored on a hard disk.

20 Fig.69 is a structural diagram of playlist information according to the present embodiment. This playlist information is an extension of the playlist information shown Fig.16 of embodiment 1, and only the differences are described here.

Cell information (Cell#) may refer to a plurality of angles, and thus information showing the number of angles ("Number") is firstly stored in the cell information, after which is stored the individual pieces of angle information 5 (Angle#).

The angle information includes the name of the VOB ("VOBName") as well as start ("In") and end ("Out") time information relating to the VOB.

Fig.70 shows an address management of angle blocks. 10 Each VOB is divided into a plurality of units (interleaved units), and the units of the VOB are interleaved stored with the units of other VOBs forming the multi-angle. For example, the data in VOB1 and VOB2 may each be divided into three interleaved units (ILVUs) as follows:

15

VOB1: ILVU(1-1), ILVU(1-2), ILVU(1-3)

VOB2: ILVU(2-1), ILVU(2-2), ILVU(2-3)

Interleaved units thus formed are stored on a disc as 20 follows:

ILVU(1-1) ILVU(1-2) ILVU(1-3) ILVU(2-1) ILVU(2-2)  
ILVU(2-3)

As shown in Fig.70, each ILVU is managed as an "extent" in the file system.

An "extent" in a file system is information showing a single continuous storage area. When there are no VOBs interleaved as in the case of a multi-angle interval, each VOB can be stored in a single continuous storage area. In this case, only a single extent is entered into an extent list, and this extent shows the start address (logical address on disc) and a data length.

10 However, in the case of a multi-angle VOB, each interleaved unit (ILVU) is contained within a single continuous storage area, and thus a single extent corresponds to a single interleaved unit.

15 Fig.71 is a structural diagram showing VOB management information according to embodiment 5. This VOB management information is an extension of the VOB management information shown in Fig.13 of embodiment 1, and only the differences will be described here.

20 In addition to stream attribute information ("Attribute") and a time map ("TMAP"), the VOB management information in the present embodiment includes an interleaved unit map ("ILVUMAP"). The interleaved unit map shows the number of interleaved units in the VOB ("Number"), as well as individual pieces of interleaved unit information

(ILVU#).

The interleaved unit information shows the number of VOBUS in the ILVU. This information can also be used with respect to non-interleaved stored VOB data generated for a 5 virtual package. Interleaved units are the switching unit in a multi-angle, and thus even if VOB data is not stored, it is still required to provide information that functions as a switching unit when a multi-angle is formed.

Fig.72 is a flowchart of event processing according to 10 embodiment 5.

Steps S6101 to S6108 correspond to steps S601 to S608 in Fig.29 of embodiment 1, and will not be described here.

The Fig.72 flowchart differs in that if judged in step S6106 that the request from the user is not a user event, 15 the processing moves to steps S6109 and below, instead of returning to step S6102.

In step S6109, the player judges whether the request from the user is to switch angles. If "yes", in step S6110 the player sums SPRM(3) (system parameter managing the angle 20 number). Next, the player judges whether the angle number shown in SPRM(3) exceeds the actual number of angles (S6111), and if "yes" SPRM(3) is reset (S6112).

If the request is not to switch angles (S6109 = "no"), or if the angle shown in SPRM(3) does actually exist (S6109

= "no"), or if step S6112 is completed, the processing returns to step S6102.

Fig.73 is a flowchart of VOB playback processing according to the present embodiment.

5 When VOB playback is started (S13001), in step S13002 information relating to the target angle is obtained from SPRM(3). Next, the player judges whether the target angle exists on the BD disc (S13003), and if "yes" the player calculates the ILVU address of the target angle (S13004),  
10 and starts reading the ILVU. On the other hand, if judged in step S13003 that the target angle is not on the BD disc (i.e. VOB data relating to the angle is on the hard disk), the player calculates the virtual ILVU address (S13005), and starts reading the ILVU. Here, in the case of a  
15 non-interleaved stored VOB, a "virtual ILVU" is an interleaved unit identified by an interleaved unit map in the VOB management information as described above.

After completing step S13004 or S13005, the player judges whether the playlist playback has ended (S13006). If 20 "no" the player judges whether reading of the target interleaved unit has ended (S13007), and if "no" the processing returns to step S13006. If judged that reading of the target interleaved unit has ended (S13007 = "yes"), the processing returns to step S13002.

Embodiment 6

An embodiment 6 of the present invention will now be described.

5 Embodiment 6 relates to the handling of a plurality of virtual packages via a home network or the like. Embodiment 6 is based on embodiments 1 to 5, and the following description focuses on extensions to and differences with embodiments 1 to 5.

10 Embodiment 6 is an example of audio data and video data realized by streaming.

Fig.74 is a functional block diagram showing AV data processing conducted by a player for playing a virtual package according to embodiment 6.

15 Fig.74 is an extension of Fig.47 in embodiment 2, and only the differences will be described here.

The player in the present embodiment is connected to the Internet 4301, and additionally includes a first streaming buffer 4302 for image data, and a second streaming buffer 4303 for audio data. Image or audio data obtained directly from the Internet as streaming is temporarily stored in these two streaming buffer, respectively.

Fig.75 is a structural diagram of VOB management information according to embodiment 6.

The VOB management information in embodiment 6 includes audio attribute information (Audio#), and the audio attribute information includes an URI showing where the streaming data is stored.

5 Fig.76 is a structural diagram of playlist information according to embodiment 6.

The playlist information in embodiment 6 includes a subtitle table ("SubtitleTable"), which in turn includes individual pieces of language information (Language#), and 10 in the language information is an URI showing where the streaming data is stored.

Fig.77 is a flowchart of AV playback processing of a virtual package according to embodiment 6.

Steps S4101 to S4107 are the same as steps 4001 to S4007 15 in Fig.52 of embodiment 2, and will not be described here.

In step S4107, the player judges whether the playlist currently being played contains audio data to be played in synchronization with the VOB being read. If "yes", the player judges whether the audio data is on the hard disk (S4108). 20 If "yes", the player starts reading, from the hard disk, ZZZ.VOBI (i.e. VOB information relating to the audio data) in step S4109, and ZZZ.VOB (i.e. audio VOB data) in step S4110, and starts sync-playback processing of the VOBs (S4111).

On the other hand, if judged that the audio data is on

the Internet (S4108 = "no"), the player starts reading NNN.VOBI (i.e. VOB information relating to the audio data) from the Internet in step S4113, and NNN.VOB (i.e. audio VOB data) from the server in step S4114, and starts streaming 5 sync-playback processing (S4115).

Figs.78 and 79 are flowcharts of VOB sync-playback processing of a stream.

The processing shown in Figs.78 and 79 is substantially the same as that shown in Figs.53 and 54 of embodiment 2, 10 respectively.

When streaming sync-playback processing is started (S9201), the player starts reading YYY.VOB (i.e. VOB file containing main video data) into a track buffer (S9202), and at the same time starts reading NNN.VOB (i.e. audio-only VOB 15 file) from the Internet into a streaming buffer (S9203). The player waits until the track buffer is full (S9204), and when full, the player judges whether there is sufficient data stored in the streaming buffer (S9205). If "yes", the player starts playback of YYY.VOB (S9206). Since the processing in 20 steps S9206 and below is the same as that in steps S9005 and below in Fig.53 of embodiment 2, description is omitted here.

Also, since the processing in steps S9301 to S9306 in Fig.79 is the same as that in steps S9201 to S9206 in Fig.78, description is omitted here. Furthermore, since the

processing in steps S9207 and below is the same as that in steps S9106 and below in Fig.54 of embodiment 2, description is omitted here.

5 Embodiment 7

An embodiment 7 of the present invention will now be described.

Embodiment 7 relates to the handling of a plurality of virtual packages via a home network or the like. Embodiment 10 7 is based on embodiments 1 to 6, and the following description focuses on extensions to and differences with embodiments 1 to 6.

Embodiment 7 relates to a conventional DVD player.

As shown in Fig.80, the playback of contents downloaded 15 from the Internet is not limited to a BD disc, but can also be conducted in conjunction with a DVD disc.

The playback of BD contents, downloaded from the Internet onto a hard disk or a memory card, in conjunction with contents on a DVD disc can be realized in the same manner 20 described in embodiments 2 to 6 for a BD disc.

Fig.81 shows an exemplary structure in the case of DVD.

As shown in Fig.81, AV data is stored on a DVD disc as 25 DVD data. Also stored on the DVD disc is the usual navigation data relating to the DVD data, as well as navigation data

relating to BD data stored on a hard disk or a memory card. Thus, the applications for a BD disc can also be realized with respect to a DVD disc.

5 Related Matters

Here, in the present invention (i.e. all of the above embodiments), the description relates to one of BD and HD-DVD. However, the present invention is not of course limited to these physical media, and can thus be applied in the case 10 of conventional DVD, CD or the like. Moreover, the present invention is not limited to physical media.

Furthermore, some of the above embodiments are described in terms of the virtual package being stored on a hard disk. However, the virtual package may instead be 15 stored on a memory card, an internal flash memory, or the like. Moreover, the present invention is not limited in terms of the recording medium that may be used for storing a virtual package.

Furthermore, in the embodiments of the present 20 invention, information relating to a virtual package is stored on a hard disk or a memory card, or obtained from the Internet, rather than being recorded onto a BD disc or the like. However, the same effects as described in the embodiments may be achieved by generating a directory on the

BD disc or the like for storing this information.

Further Variations

The present invention is based on the embodiments as 5 described above. However, the present invention is, of course, not limited to these embodiments, and the following variations may also be included in the present invention.

(1) The various apparatuses described above form a computer 10 system structured from a microprocessor, a ROM, a RAM, and the like. Computer programs are stored on the RAM or a hard disk unit. The various devices function as a result of the microprocessor performing operation in accordance with the computer programs.

15

(2) The present invention may be the methods described above. Alternatively, the present invention may be a computer program realized by a computer, or a digital signal expressing the computer program.

20 Furthermore, the present invention may be a computer readable recording medium for recording the computer program or the digital signal, examples of which include a flexible disk, a hard disk, a CD-ROM, an MO, a DVD, a DVD-ROM, a DVD-RAM, a BD (blu-ray disc), a semiconductor memory, or the like.

Moreover, the present invention may be the computer program or the digital signal stored on any of the above recording media.

Furthermore, the present invention may be the computer program or the digital signal transmitted via a network or the like, such as an electronic communications circuit, a radio communications circuit, a cable communications circuit, and the Internet.

Furthermore, the present invention may be a computer system that includes a microprocessor and a memory, the computer program may be stored in the memory, and the microprocessor may operate in accordance with the computer program.

Furthermore, the computer program or the digital signal may be realized in another computer system by transferring the recording medium on which the computer program or the digital signal is stored to the other computer system, or by transferring the computer program or the digital signal to the other computer system via the network or the like.

20

(3) According to the present invention, any combination of the above embodiments and variations is possible.

### Effects of the Invention

In the present invention, as a result of a BD player being organically connected to a hard disk or a memory card, original contents stored on a BD disc can be supplemented with video, audio, and subtitle data at a later date. Consequently, the original contents can be updated in a way that is not possible according to conventional package media.

### List of Symbols

10	201	BD disc
	202	optical pickup
	203	program storage memory
	204	management information storage memory
	205	AV storage memory
15	206	program processing unit
	207	management information processing unit
	208	presentation processing unit
	209	image plane
	210	video plane
20	211	integration processing unit
	2001	hard disk
	2002	reading control unit
	2003	virtual package storage unit
	2004	virtual package generation unit

2005 virtual package display control unit  
2006 virtual package table

301 program storage memory  
5 302 program processor  
303 UOP manager  
304 management information storage memory  
305 scenario processor  
306 presentation controller

10 307 clock  
308 image memory  
309 track buffer  
310 demultiplexer  
311 image processor

15 312 video processor  
313 audio processor  
314 image plane  
315 video plane  
316 integration processing unit

20 317 drive controller

4001 hard disk  
4002 BD disc  
4003 track buffer

4004 audio buffer  
4005 first image memory  
4006 second image memory  
4007 demultiplexer  
5 4008 image processor  
4009 video processor  
4010 audio processor  
4011 clock  
4012 adder/switcher  
10 4101 first buffer  
4102 second buffer  
4103 filter  
4104 demultiplexer core  
4201 offset  
15 4202 first demultiplexer core  
4203 second demultiplexer core  
4204 first audio processor core  
4205 second audio processor core  
4206 adder/switcher  
20 4301 Internet  
4302 first streaming buffer  
4303 second streaming buffer  
  
S101 disc inserting step

S102 BD.INFO reading step  
S103 BD.PROG reading step  
S104 first event generating step  
S105 event handler executing step

5  
S201 UOP receiving step  
S202 UOP event generating step  
S203 menu call judging step  
S204 event generating step  
10 S205 event handler executing step

S301 playlist playback starting step  
S302 playlist information (XXX.PL) reading step  
S303 playlist program (XXX.PROG) reading step  
15 S304 cell playback starting step  
S305 AV playback starting step

S401 AV playback starting step  
S402 VOB information (YYY.VOBI) reading step  
20 S403 VOB (YYY.VOB) reading step  
S404 VOB playback starting step  
S405 VOB playback ending step  
S406 next cell judging step

5 S501 playlist playback starting step  
S502 playlist playback end judging step  
S503 time event time judging step  
S504 event generating step  
5 S505 event handler executing step

10 S601 playlist playback starting step  
S602 playlist playback end judging step  
S603 UOP receipt judging step  
S604 UOP event generating step  
S605 menu call judging step  
S606 user event valid period judging step  
S607 event generating step  
S608 event handler executing step

15 S701 playlist playback starting step  
S702 playlist playback end judging step  
S703 subtitle draw start judging step  
S704 subtitle drawing step  
20 S705 subtitle display end judging step  
S706 subtitle erasing step

S1001 virtual package generating step  
S1002 BD.INFO reading step

S1003 BD.PROG reading step  
S1004 first event generating step  
S1005 event handler executing step

5 S1101 virtual package generating step  
S1102 BD.INFO reading step  
S1103 BD.PROG reading step  
S1104 end position information-based  
playback starting step

10 S2001 UOP receiving step  
S2002 UOP event generating step  
S2003 menu call judging step  
S2004 event generating step  
15 S2005 event handler executing step

S3001 PL playback starting step  
S3002 PL storage position judging step  
S3003 playlist exchanging step

20 S3004 playlist storage position judging step  
S3005 HDD playlist information (XXX.PL) reading step  
S3006 HDD playlist information (XXX.PROG) reading step  
S3007 BD playlist information (XXX.PL) reading step  
S3008 BD playlist information (XXX.PROG) reading step

S3009 cell playback starting step  
S3010 AV playback starting step

S4001 AV playback starting step  
5 S4002 VOB storage position judging step  
S4003 HDD VOB information (YYY.VOBI) reading step  
S4004 HDD VOB (YYY.VOB) reading step  
S4005 BD VOB information (YYY.VOBI) reading step  
S4006 BD VOB (YYY.VOB) reading step  
10 S4007 supplementary audio judging step  
S4008 HDD VOB information (ZZZ.VOBI) reading step  
S4009 HDD VOB (ZZZ.VOB) reading step  
S4010 VOB sync processing starting step  
S4011 VOB playback starting step

15 S4101 AV playback starting step  
S4102 VOB storage position judging step  
S4103 HDD VOB information (YYY.VOBI) reading step  
S4104 HDD VOB (YYY.VOB) reading step  
20 S4105 BD VOB information (YYY.VOBI) reading step  
S4106 BD VOB (YYY.VOB) reading step  
S4107 supplementary audio judging step  
S4108 HDD supplementary audio judging step  
S4109 HDD VOB information (ZZZ.VOBI) reading step

S4110           HDD VOB (ZZZ.VOB) reading step  
S4111           VOB sync processing starting step  
S4112           VOB playback starting step  
S4113           VOB information (ZZZ.VOBI) reading step  
5    S4114           server VOB (NNN.VOB) reading step  
S4115           streaming sync processing starting step

S6001           playlist playback starting step  
S6002           playlist playback end judging step  
10   S6003           UOP receiving step  
S6004           UOP event generating step  
S6005           menu call judging step  
S6006           user event valid period judging step  
S6007           event generating step  
15   S6008           event handler executing step  
S6009           audio switching judging step  
S6010           system parameter SPRM(1) summing step  
S6011           corresponding audio judging step  
S6012           system parameter SPRM(1) resetting step  
20   S6013           subtitle switching judging step  
S6014           system parameter SPRM(2) summing step  
S6015           corresponding subtitle judging step  
S6016           system parameter SPRM(2) resetting step

S6101       playlist playback starting step  
S6102       playlist playback end judging step  
S6103       UOP receiving step  
S6104       UOP event generating step  
5    S6105       menu call judging step  
S6106       user event valid period judging step  
S6107       event generating step  
S6108       event handler executing step  
S6109       angle switching judging step  
10   S6110       system parameter SPRM(3) supplementing step  
S6111       corresponding angle judging step  
S6112       system parameter SPRM(3) resetting step

S7001       playlist playback starting step  
15   S7002       playlist playback end judging step  
S7003       draw interval end PNG judging step  
S7004       PNG discarding step  
S7005       subtitle draw start judging step  
S7006       subtitle drawing step  
20   S7007       subtitle switching judging step  
S7008       display subtitle judging step  
S7009       subtitle erasing step  
S7010       display valid subtitle judging step  
S7011       subtitle drawing step

S7012 subtitle end judging step  
S7013 subtitle erasing step

5 S8001 disc inserting step  
S8002 URI obtaining step  
S8003 server BD.INFO judging step  
S8004 server BD.INFO obtaining step  
S8005 new information confirming step  
S8006 contents downloading step

10 S8007 HDD BD.INFO judging step  
S8008 virtual package playing step  
S8009 BD playing step

S9001 VOB sync processing starting step  
15 S9002 VOB file (YYY.VOB) read starting step  
S9003 audio VOB file (ZZZ.VOB) read starting step  
S9004 track buffer storage amount judging step  
S9005 VOB file (YYY.VOB) playback starting step  
S9006 sync timing standby step

20 S9007 audio VOB file (ZZZ.VOB) playback starting  
step  
S9008 playback audio judging step  
S9009 SCR match judging step  
S9010 packet exchanging step

S9011        VOB playback end judging step  
S9012        next cell judging step  
S9013        cell playback starting step

5    S9101        VOB sync processing starting step  
S9102        VOB file (YYY.VOB) read starting step  
S9103        audio VOB file (ZZZ.VOB) read starting step  
S9104        track buffer storage amount judging step  
S9105        VOB file (YYY.VOB) playback starting step

10   S9106        sync timing standby step  
S9107        audio VOB file (ZZZ.VOB) playback starting  
              step  
S9108        playback audio judging step  
S9109        VOB file (YYY.VOB) audio playing step

15   S9110        audio VOB file (ZZZ.VOB) audio playing step  
S9111        VOB playback end judging step  
S9112        next cell judging step  
S9113        cell playback starting step

20   S9201        VOB sync processing starting step  
S9202        VOB file (YYY.VOB) read starting step  
S9203        audio VOB file (NNN.VOB) read starting step  
S9204        track buffer storage amount judging step  
S9205        streaming buffer ready judging step

S9206 VOB file (YYY.VOB) playback starting step  
S9207 sync timing standby step  
S9208 audio VOB file (NNN.VOB) playback starting  
step  
5 S9209 playback audio judging step  
S9210 SCR match judging step  
S9211 packet exchanging step  
S9212 VOB playback end judging step  
S9213 next cell judging step  
10 S9214 cell playback starting step  
  
S9301 VOB sync processing starting step  
S9302 VOB file (YYY.VOB) read starting step  
S9303 audio VOB file (NNN.VOB) read starting step  
15 S9304 track buffer storage amount judging step  
S9305 streaming buffer ready judging step  
S9306 VOB file (YYY.VOB) playback starting step  
S9307 sync timing standby step  
S9308 audio VOB file (NNN.VOB) playback starting  
step  
20 S9309 playback audio judging step  
S9310 VOB file (YYY.VOB) audio playing step  
S9311 audio VOB file (NNN.VOB) audio playing step  
S9312 VOB playback end judging step

S9313        next cell judging step  
S9314        cell playback starting step

S10001        disc inserting step  
5    S10002        virtual package generating step  
S10003        virtual package information displaying step  
S10004        user inputting step  
S10005        selected virtual package information storing  
              step

10   S10006        virtual package playing step

S11001        disc inserting step  
S11002        virtual package information judging step  
S11003        storage position information obtaining step  
15   S11004        storage position information judging step  
S11005        BD.INFO obtaining step  
S11006        virtual package playing step  
S11007        BD playback ending step  
S11008        end position information updating step

20   S11009        normal BD playing step  
S11010        BD playback ending step  
S11011        end position information updating step  
S11012        normal BD playing step  
S11013        BD playback ending step

S11014 end position information generating step

S12001 disc inserting step

S12002 local HDD connection judging step

5 S12003 HDD virtual package information judging step

S12004 server connection judging step

S12005 server virtual package information judging step

S12006 memory card connection judging step

S12007 memory card virtual package information judging

10 step

S12008 virtual package information reading step

S12009 virtual package playing step

S12010 normal BD playing step

15 S13001 PL playback starting step

S13002 target angle obtaining step

S13003 target angle storage position judging step

S13004 ILVU calculating step

S13005 virtual ILVU calculating step

20 S13006 PL playback end judging step

S13007 ILVU reading step

Although the present invention has been fully described by way of examples with reference to the accompanying

drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Therefore, unless such changes and modifications depart from the scope of the present invention, they should be construed as being included therein.

WHAT IS CLAIMED IS:

1 1. A playback apparatus for playing back an optical disc that  
2 stores at least video data and audio data, comprising:  
3 a storage unit that is at least one of a magnetic storage  
4 device and a removable nonvolatile memory, wherein  
5 the storage unit stores at least one of audio data and  
6 subtitle data, and  
7 the playback apparatus plays back (i) at least one piece  
8 of each of the video and audio data stored on the optical  
9 disc in synchronization with (ii) at least one piece of the  
10 audio or subtitle data stored in the storage unit.

1 2. The playback apparatus of claim 1, wherein  
2 the storage unit stores sync-playback management  
3 information relating to the audio or subtitle data, stored  
4 in the storage unit, that is to be played back in  
5 synchronization with the video data stored on the optical  
6 disc, and  
7 the playback apparatus comprises units operable to read  
8 the sync-playback management information; and perform, in  
9 accordance with the sync-playback management information,  
10 sync-playback processing of (i) the video data, stored on  
11 the optical disc, that is to be synchronously played back,  
12 and (ii) the audio or subtitle data, stored in the storage

13 unit, that is to be synchronously played back.

1 3. The playback apparatus as in one of claims 1 and 2, wherein  
2 the video and audio data on the optical disc is stored  
3 in a multiplexed state,

4 (i) a timestamp of the audio data multiplexed on the  
5 optical disc with the video data to be synchronously played  
6 back matches (ii) a timestamp of the audio data, stored in  
7 the storage unit, that is to be synchronously played back,  
8 and

9 the playback apparatus comprises a unit operable to  
10 compare (i) the timestamp of the audio data multiplexed on  
11 the optical disc with the video data to be synchronously  
12 played back with (ii) the timestamp of the audio data, stored  
13 in the storage unit, that is to be synchronously played back.

1 4. A playback apparatus for playing back an optical disc that  
2 stores at least video data and audio data, comprising:

3 a storage unit that is at least one of a magnetic storage  
4 device and a removable nonvolatile memory, wherein  
5 the storage unit stores at least video data and audio  
6 data, and

7 the playback apparatus plays back, in accordance with  
8 playback-sequence management information stored in the

9 storage unit, the video and audio data stored on the optical  
10 disc, and the video and audio data stored in the storage unit.

1 5. A playback apparatus for playing back an optical disc that  
2 stores at least video data and audio data, comprising:  
3 a storage unit that is at least one of a magnetic storage  
4 device and a removable nonvolatile memory, wherein  
5 the storage unit stores at least video data and audio  
6 data,

7 the video and audio data stored on the optical disc  
8 includes, when playback is to be conducted by the playback  
9 apparatus, angle intervals for dynamically switching  
10 playback scenes, and

11 the playback apparatus plays back, as one of the angle  
12 intervals, at least one piece of the video data stored in  
13 the storage unit.

1 6. The playback apparatus as in any of claims 1 to 5, wherein  
2 the optical disc stores first playback management  
3 information for regulating a playback sequence of the video  
4 and audio data stored on the optical disc,  
5 the storage unit stores second playback management  
6 information for regulating a playback sequence of (i) the  
7 video and audio data stored on the optical disc and (ii) the

8 video and audio data stored in the storage unit, and  
9 the playback apparatus generates third playback  
10 management information based on the first and second playback  
11 management information.

1 7. The playback apparatus of claim 6, wherein  
2 a part or all of the first playback management  
3 information having the same identification information as  
4 the second playback management information is nullified by  
5 the second playback management information.

1 8. The playback apparatus as in any of claims 1 to 7, wherein  
2 the storage unit stores, for each optical disc, (i)  
3 audio or subtitle data to be played back in synchronization  
4 with video and audio data stored on the optical disc, and  
5 (ii) video and audio data to be played back, in a predetermined  
6 playback sequence, with video and audio data stored on the  
7 optical disc, and

8 the playback apparatus stores a table for managing, for  
9 each optical disc, (i) the audio or subtitle data, stored  
10 in the storage unit, that is to be synchronously played back,  
11 and (ii) the video and audio data, stored in the storage unit,  
12 that is to be played back in the predetermined playback  
13 sequence.

1 9. The playback apparatus as in any of claims 1 to 8, wherein  
2 the playback apparatus is connected to a network,  
3 another playback apparatus is connected to the network,  
4 and  
5 the other playback apparatus refers, at a time of  
6 playing back an optical disc, to the table stored in the  
7 playback apparatus.

1 10. The playback apparatus as in any of claims 1 to 9, wherein  
2 the playback apparatus:  
3 includes an interface for connecting to the Internet,  
4 and  
5 obtains, from the Internet, audio data to be played back  
6 in synchronization with the video data stored on the optical  
7 disc.

1 11. A playback apparatus for playing back an optical disc  
2 that stores at least video data and audio data, comprising:  
3 a removable nonvolatile memory, wherein  
4 the removable nonvolatile memory stores playback  
5 control information for managing the playback of the video  
6 and audio data stored on the optical disc, and  
7 the playback apparatus reads the playback control  
8 information from the removable nonvolatile memory, and plays

9 back, in accordance with the playback control information,  
10 the video and audio data stored on the optical disc.

1 12. A playback method in a playback system for playing back  
2 an optical disc that stores at least video data and audio  
3 data, the playback system including a storage unit that is  
4 at least one of a magnetic storage unit and a removable  
5 nonvolatile memory, and the storage unit storing at least  
6 one of audio data and subtitle data, the playback method  
7 comprising the step of:

8 playing back (i) at least one piece of each of the video  
9 and audio data stored on the optical disc in synchronization  
10 with (ii) at least one piece of the audio or subtitle data  
11 stored in the storage unit.

1 13. The playback method of claim 12, wherein  
2 the storage unit stores sync-playback management  
3 information relating to the audio or subtitle data, stored  
4 in the storage unit, that is to be played back in  
5 synchronization with the video data stored on the optical  
6 disc, and

7 the playback method comprises the steps of reading the  
8 sync-playback management information; and performing, in  
9 accordance with the sync-playback management information,

10 sync-playback processing of (i) the video data, stored on  
11 the optical disc, that is to be synchronously played back,  
12 and (ii) the audio or subtitle data, stored in the storage  
13 unit, that is to be synchronously played back.

1 14. The playback method as in one of claims 12 and 13, wherein  
2 the video and audio data on the optical disc is stored  
3 in a multiplexed state,

4 (i) a timestamp of the audio data multiplexed on the  
5 optical disc with the video data to be synchronously played  
6 back matches (ii) a timestamp of the audio data, stored in  
7 the storage unit, that is to be synchronously played back,  
8 and

9 the playback method comprises the step of comparing (i)  
10 the timestamp of the audio data multiplexed on the optical  
11 disc with the video data to be synchronously played back with  
12 (ii) the timestamp of the audio data, stored in the storage  
13 unit, that is to be synchronously played back.

1 15. A playback method in a playback system for playing back  
2 an optical disc that stores at least video data and audio  
3 data, the playback system including a storage unit that is  
4 at least one of a magnetic storage unit and a removable  
5 nonvolatile memory, and the storage unit storing at least

6 video data and audio data, the playback method comprising  
7 the step of:

8 playing back, in accordance with playback-sequence  
9 management information stored in the storage unit, the video  
10 and audio data stored on the optical disc and the video and  
11 audio data stored in the storage unit.

1 16. A playback method in a playback system for playing back  
2 an optical disc that stores at least video data and audio  
3 data, the playback system including a storage unit that is  
4 at least one of a magnetic storage unit and a removable  
5 nonvolatile memory, and the storage unit storing at least  
6 video data and audio data, wherein

7 the video and audio data stored on the optical disc  
8 includes, when playback is to be conducted by the playback  
9 system, angle intervals for dynamically switching playback  
10 scenes, and

11 the playback method comprises the step of playing back,  
12 as one of the angle intervals, at least one piece of the video  
13 data stored in the storage unit.

1 17. The playback method as in any of claims 12 to 16, wherein  
2 the optical disc stores first playback management  
3 information for regulating a playback sequence of the video

4 and audio data stored on the optical disc,  
5 the storage unit stores second playback management  
6 information for regulating a playback sequence of (i) the  
7 video and audio data stored on the optical disc and (ii) the  
8 video and audio data stored in the storage unit, and  
9 the playback method comprises the step of generating  
10 third playback management information based on the first and  
11 second playback management information.

1 18. The playback method of claim 17, wherein  
2 a part or all of the first playback management  
3 information having the same identification information as  
4 the second playback management information is nullified by  
5 the second playback management information.

1 19. The playback method as in any of claims 12 to 18, wherein  
2 the playback method comprises the steps of:  
3 storing in the storage unit, for each optical disc, (i)  
4 audio or subtitle data to be played back in synchronization  
5 with video and audio data stored on the optical disc, and  
6 (ii) video and audio data to be played back, in a predetermined  
7 playback sequence, with video and audio data stored on the  
8 optical disc; and  
9 storing, in the playback system, a table for managing,

10 for each optical disc, (i) the audio or subtitle data, stored  
11 in the storage unit, that is to be synchronously played back,  
12 and (ii) the video and audio data, stored in the storage unit,  
13 that is to be played back in the predetermined playback  
14 sequence.

1 20. The playback method as in any of claims 12 to 19, wherein  
2 the playback system is connected to a network,  
3 another playback system is connected to the network,  
4 and  
5 the other playback system refers, at a time of playing  
6 back an optical disc, to the table stored in the playback  
7 system.

1 21. The playback method as in any of claims 12 to 20, wherein  
2 the playback system includes an interface for  
3 connecting to the Internet, and  
4 the playback method comprises the step of obtaining,  
5 from the Internet, audio data to be played back in  
6 synchronization with the video data stored on the optical  
7 disc.

1 22. A playback method in a playback system for playing back  
2 an optical disc that stores at least video data and audio

3 data, the playback system including a removable nonvolatile  
4 memory, and the removable nonvolatile memory storing  
5 playback control information for managing the playback of  
6 the video and audio data stored on the optical disc, the  
7 playback method comprising the steps of:

8 reading the playback control information from the  
9 removable nonvolatile memory; and

10 playing back, in accordance with the playback control  
11 information, the video and audio data stored on the optical  
12 disc.

1 23. An optical disc storing at least video data and audio  
2 data, and played back by a playback apparatus, the playback  
3 apparatus including a storage unit that is at least one of  
4 a magnetic storage device and a removable nonvolatile memory,  
5 and the storage unit storing at least one of audio data and  
6 subtitle data, wherein

7 at least one piece of the video data stored on the  
8 optical disc is played back in synchronization with at least  
9 one piece of the audio or subtitle data stored in the storage  
10 unit, and

11 the optical disc stores a part or all of reference  
12 information that is for specifying the audio or subtitle data,  
13 stored in the storage unit, that is to be played back in

14 synchronization with the video data stored on the optical  
15 disc.

1 24. The optical disc of claim 23, wherein  
2 the storage unit stores sync-playback management  
3 information relating to the audio or subtitle data, stored  
4 in the storage unit, that is to be played back in  
5 synchronization with the video data stored in the optical  
6 disc, and

7 the video data to be synchronously played back and the  
8 audio or subtitle data to be synchronously played back is  
9 sync-playback processed by the playback apparatus in the  
10 accordance with the sync-playback management information.

1 25. The optical disc as in one of claims 23 and 24, wherein  
2 the video and audio data on the optical disc is stored  
3 in a multiplexed state, and  
4 (i) a timestamp of the audio data multiplexed on the  
5 optical disc with the video data to be synchronously played  
6 back matches (ii) a timestamp of the audio data, stored in  
7 the storage unit, that is to be synchronously played back.

1 26. The optical disc as in any of claims 23 to 25, wherein  
2 the optical disc and the storage unit each store

3 access-support information relating to video and audio data  
4 stored respectively,

5 the access-support information is a table for managing,  
6 in arbitrary units, time information and data size  
7 information, and

8 (i) an audio playback time period, managed in the table  
9 on the optical disc, that relates to the audio data stored  
10 on the optical disc with video data that is to be synchronously  
11 played back matches (ii) an audio playback time period,  
12 managed in the table in the storage unit, that relates to  
13 the audio data, stored in the storage unit, that is to be  
14 synchronously played back.

1 27. An optical disc storing at least video data and audio  
2 data, and played back by a playback apparatus, the playback  
3 apparatus including a storage unit that is at least one of  
4 a magnetic storage device and a removable nonvolatile memory,  
5 and the storage unit storing video data, wherein  
6 the video data stored on the optical disc is played back,  
7 in a predetermined playback sequence, with the video data  
8 stored in the storage unit,

9 the optical disc stores a part or all of reference  
10 information that is for specifying the video data, stored  
11 in the storage unit, that is to be played back, in the

12 predetermined playback sequence, with the video data stored  
13 on the optical disc.

1 28. An optical disc storing at least video data and audio  
2 data, and played back by a playback apparatus, the playback  
3 apparatus including a storage unit that is at least one of  
4 a magnetic storage device and a removable nonvolatile memory,  
5 and the storage unit storing at least video data and audio  
6 data, wherein

7 the video and audio data stored on the optical disc  
8 includes, when playback is to be conducted by the playback  
9 apparatus, multi-angle intervals for dynamically switching  
10 playback scenes,

11 at least one piece of each of the video and audio data  
12 stored in the storage unit is played back as one of the angles  
13 of the multi-angle intervals,

14 the optical disc stores a part or all of reference  
15 information that is for specifying the video and audio data,  
16 stored in the storage unit, that is to be played back as one  
17 of the angles of the multi-angle intervals.

1 29. The optical disc as in any of claims 23 to 28, wherein  
2 the playback apparatus includes an interface for  
3 connecting to the Internet, and obtains, from the Internet,

4      audio data to be played back in synchronization with the video  
5      data stored on the optical disc, and  
6               the optical disc stores access point information  
7      relating to the Internet.

## ABSTRACT

With DVD, it is difficult to create new applications that are organically linked to contents on the Internet, because of conventional recording methods in which a part 5 of navigation information is embedded in a stream. In the present invention, data for a package is structured so that the stream and the navigation information are completely separate, and also so that the independence of individual playback units is enhanced. This allows for operations to 10 be conducted without conflict, even when new streams and navigation information are supplemented to streams and navigation information existing on a disc. Consequently, data such as audio, subtitle, and video data downloaded from the Internet and stored on a hard disk, memory card or the 15 like can be readily played back in synchronization with the contents on the disc.

FIG.1

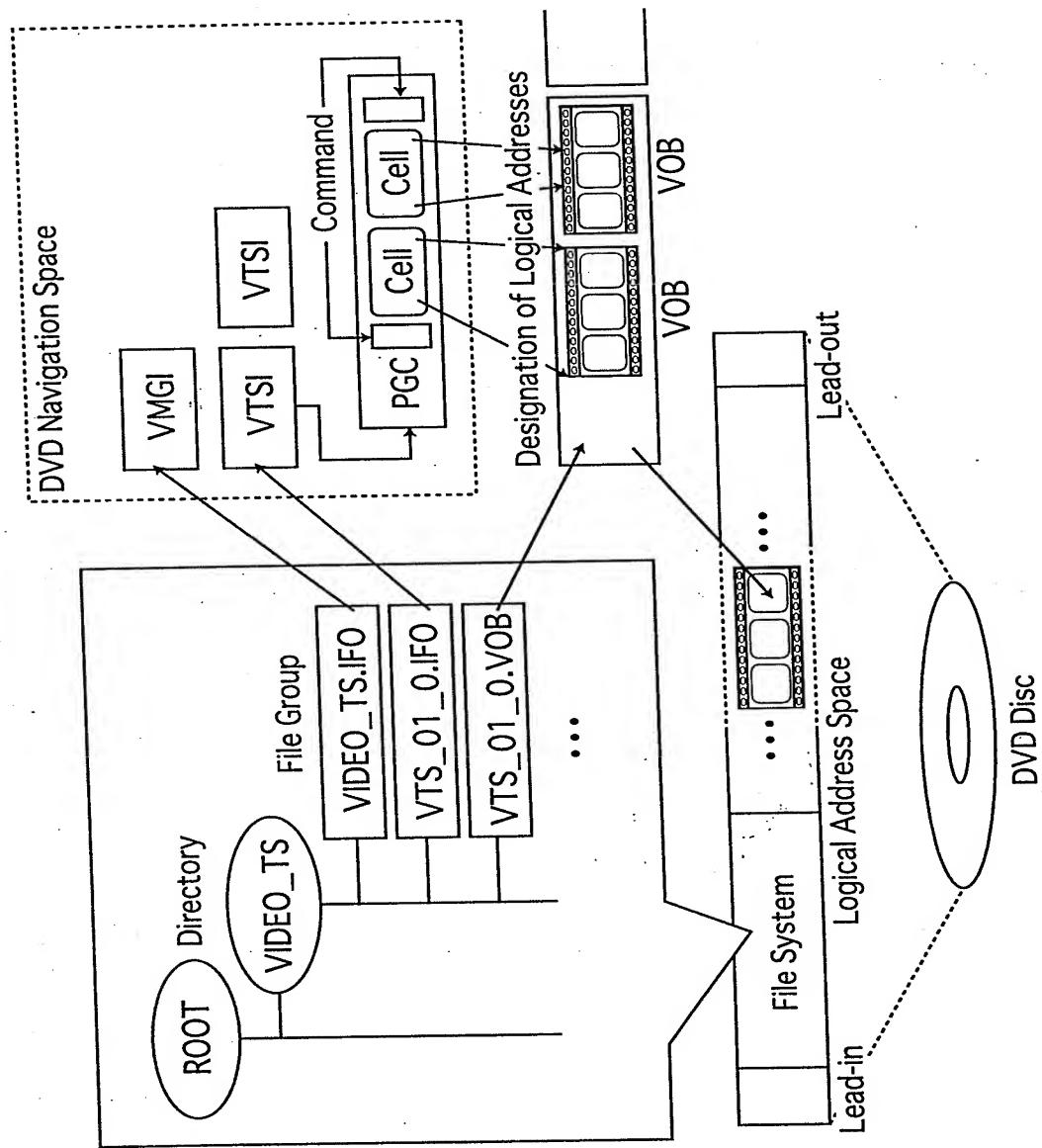


FIG.2

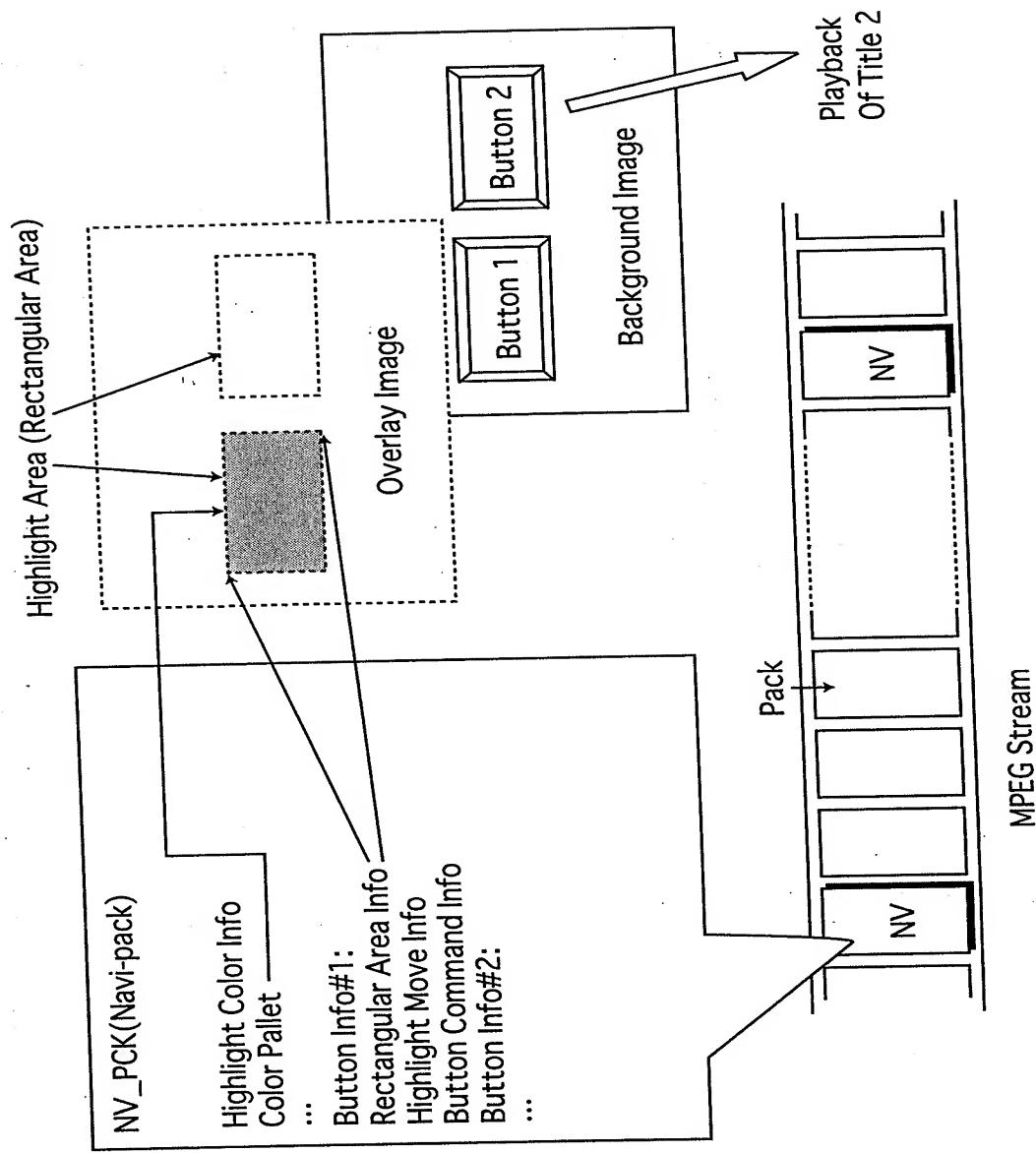


FIG.3

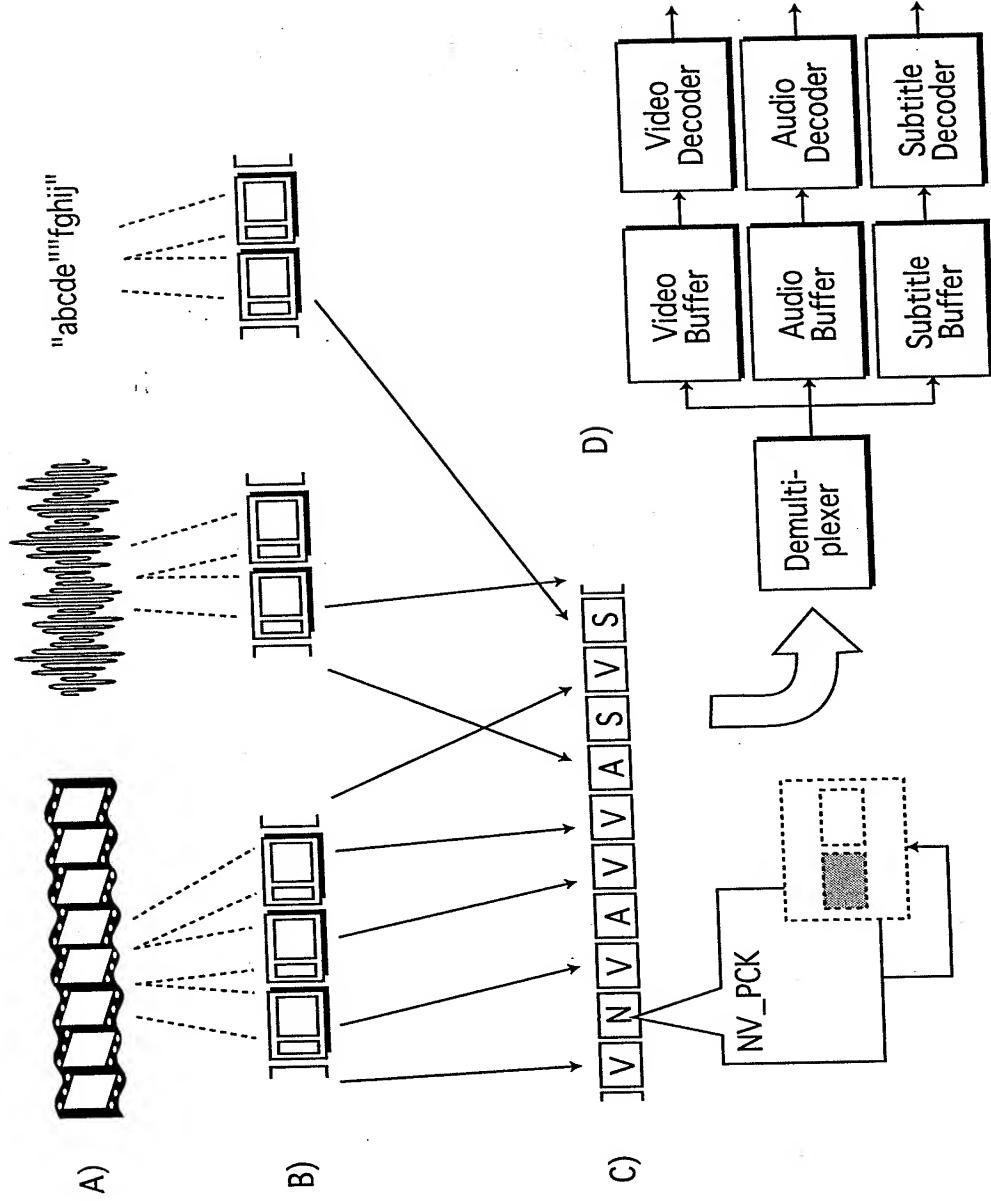


FIG.4

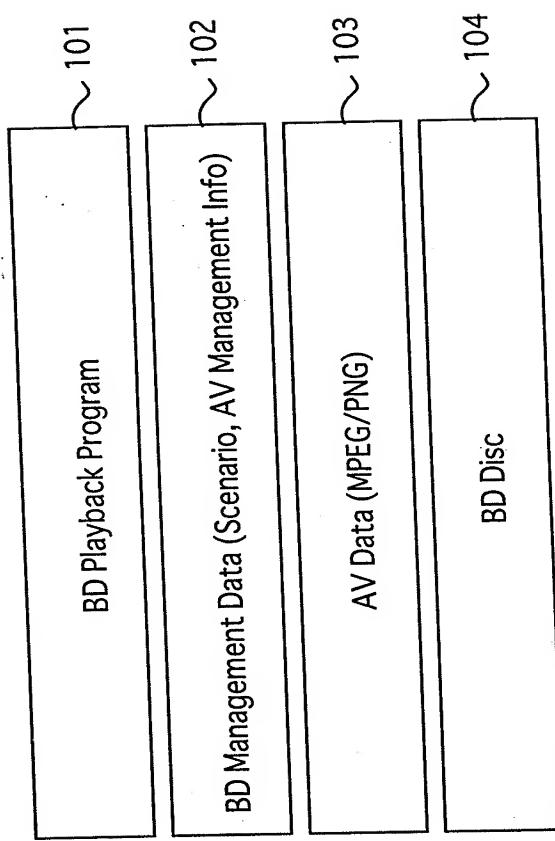


FIG.5

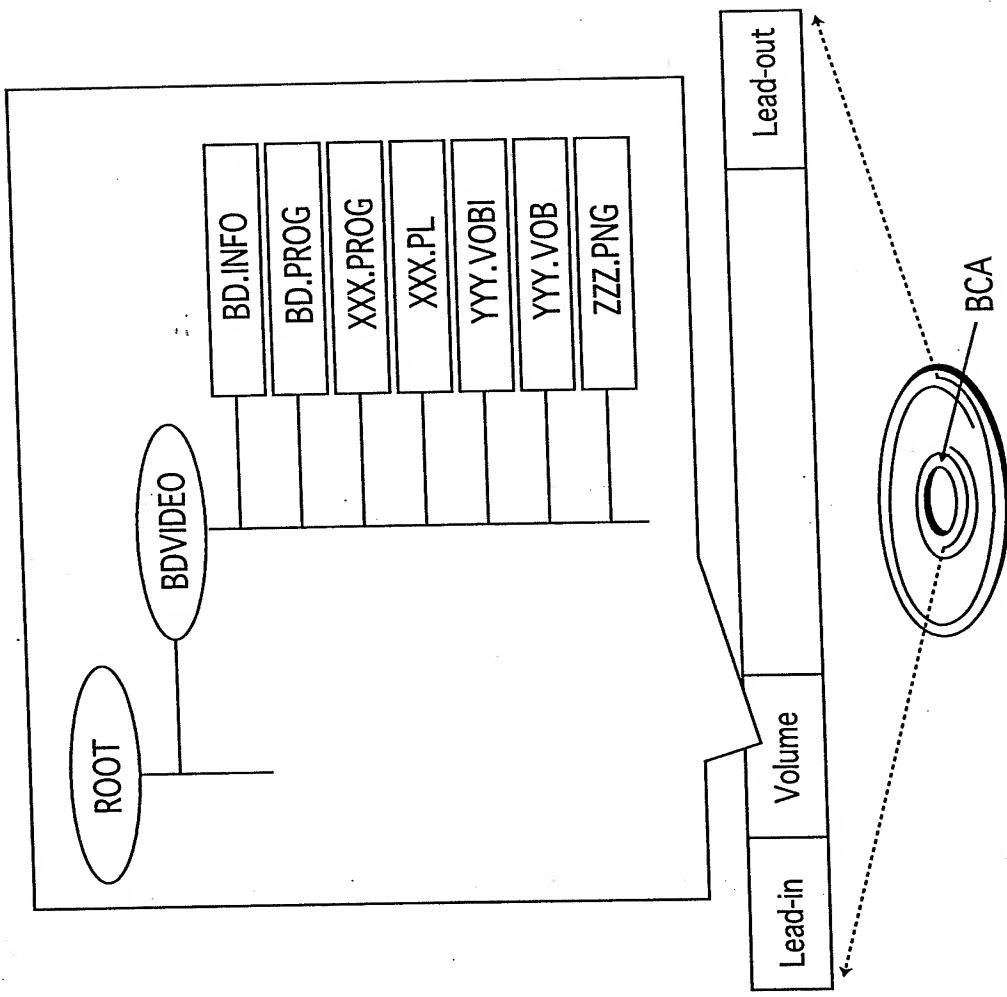


FIG.6

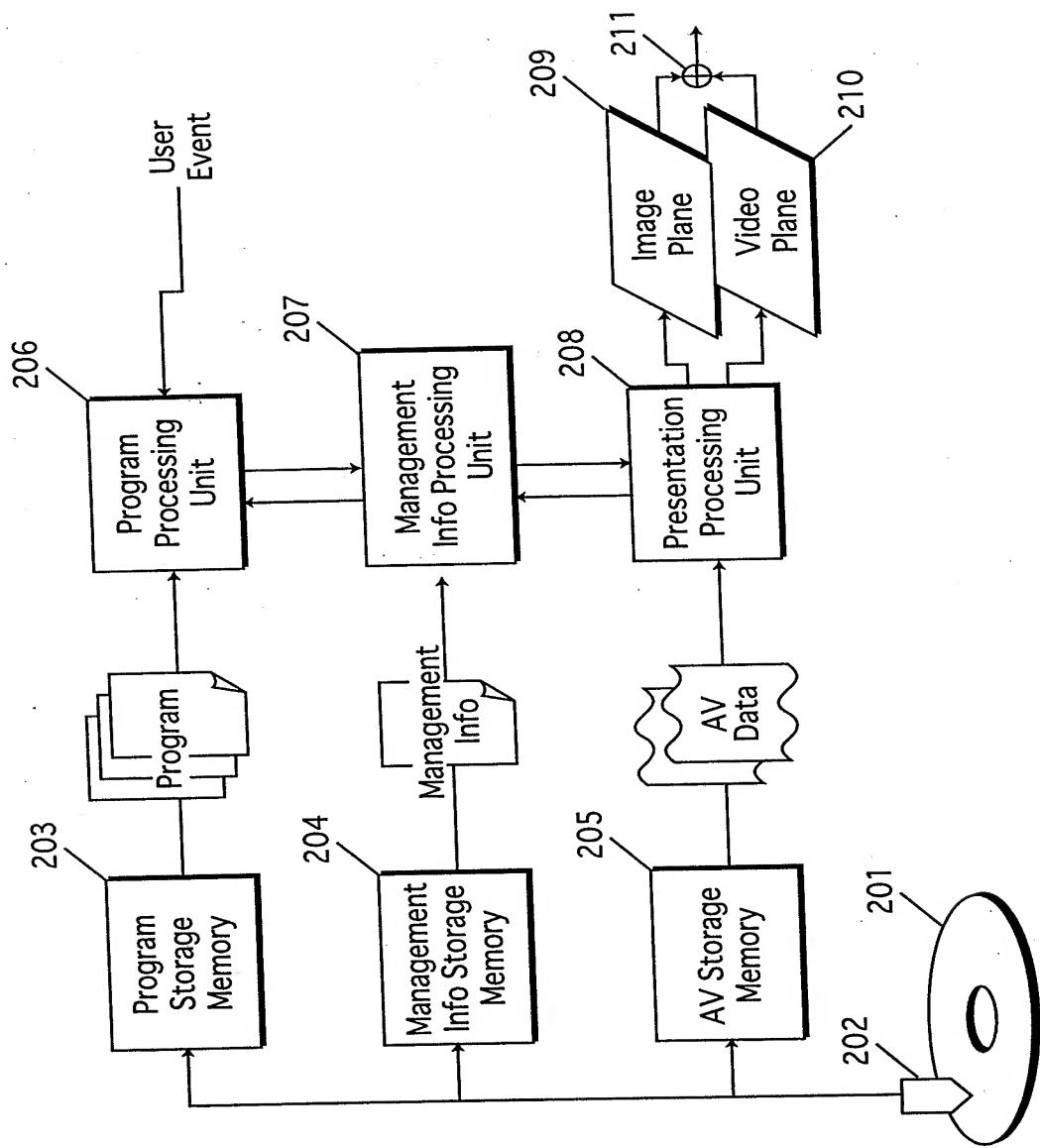


FIG.7

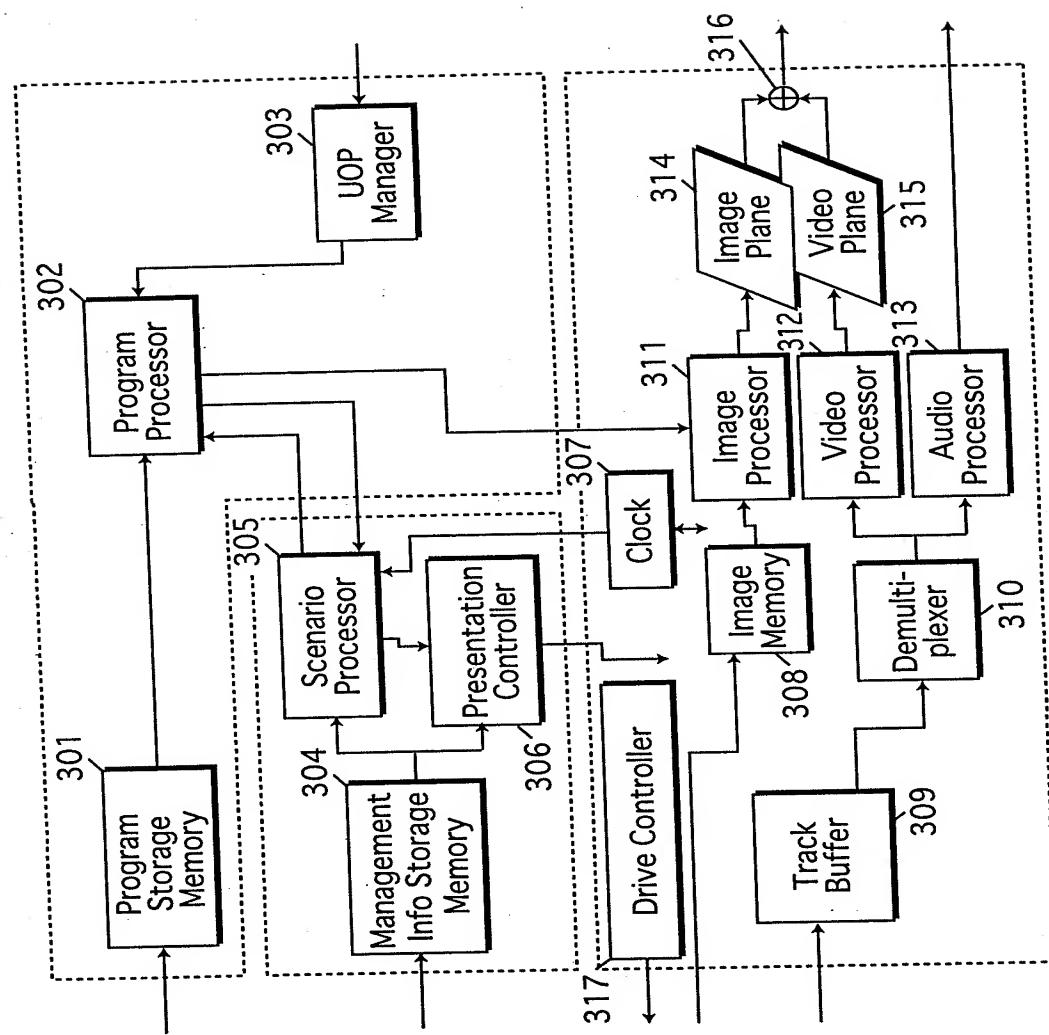


FIG.8

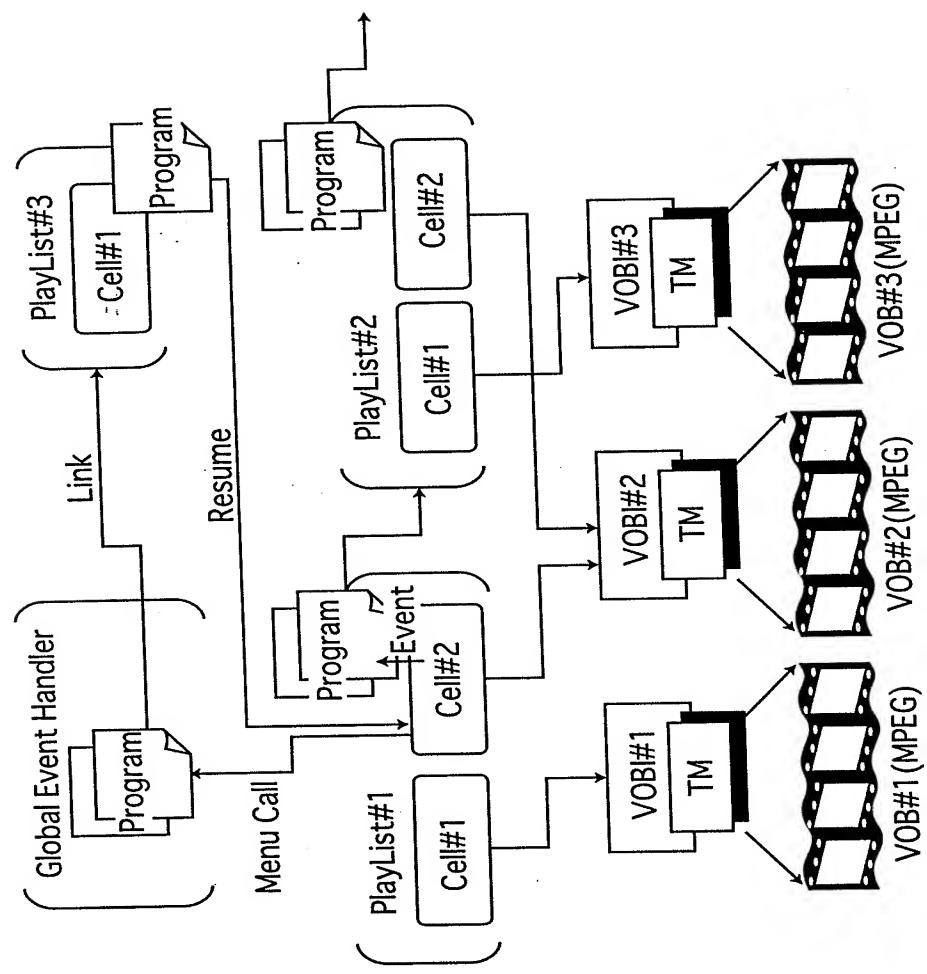


FIG.9

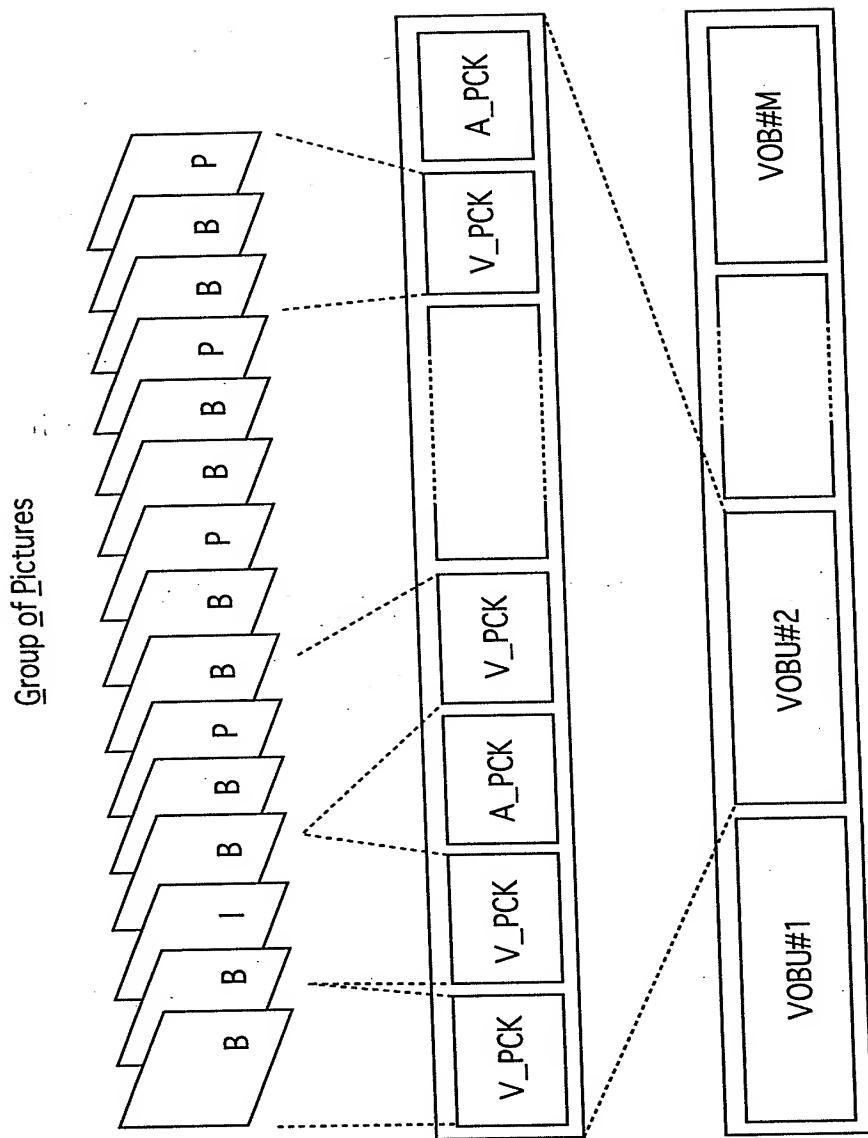


FIG.10

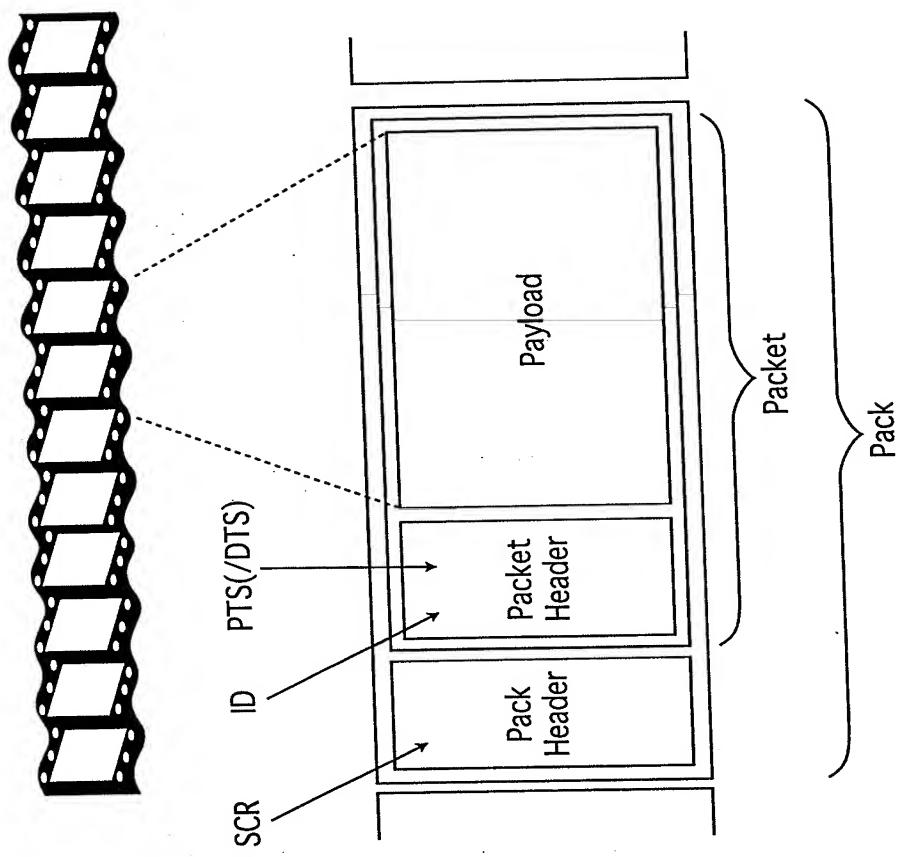


FIG. 1.1

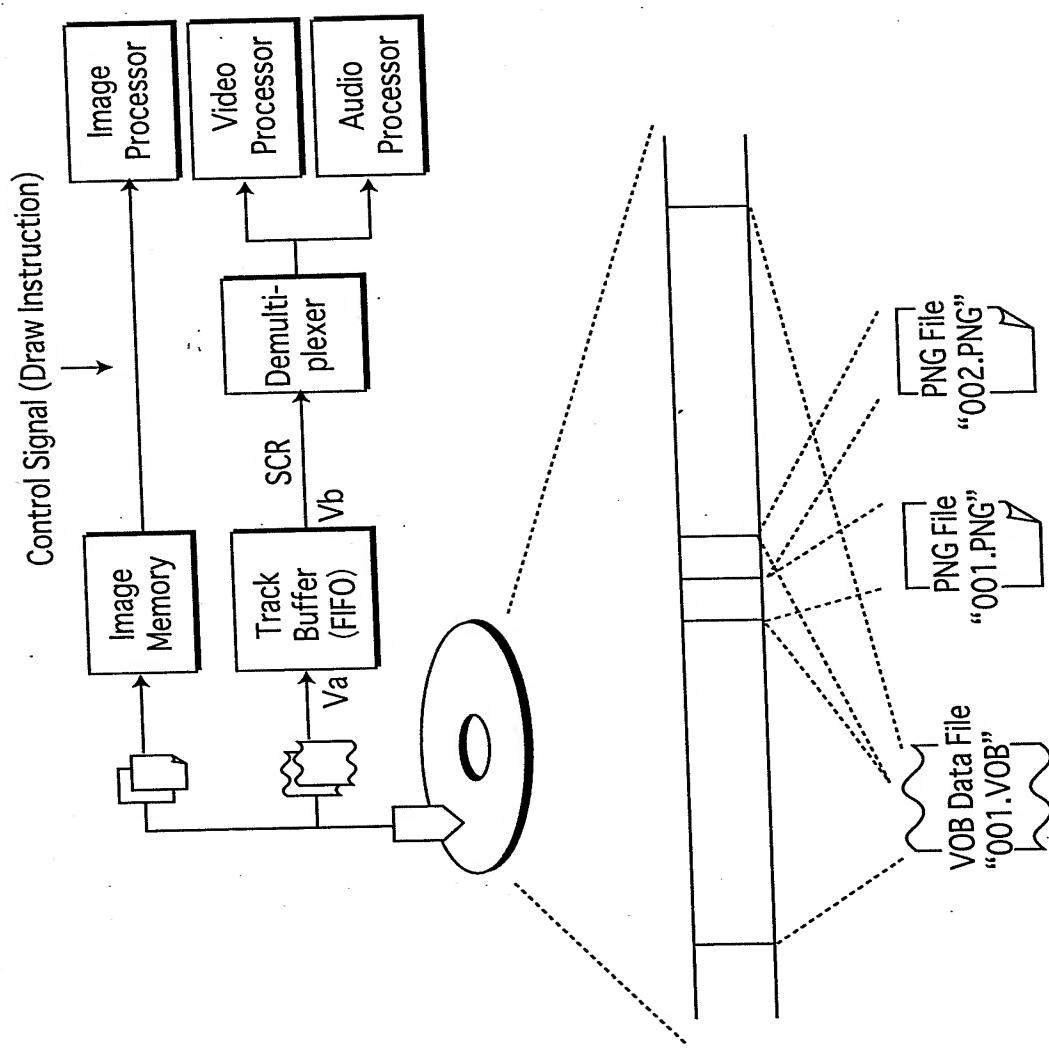


FIG.12

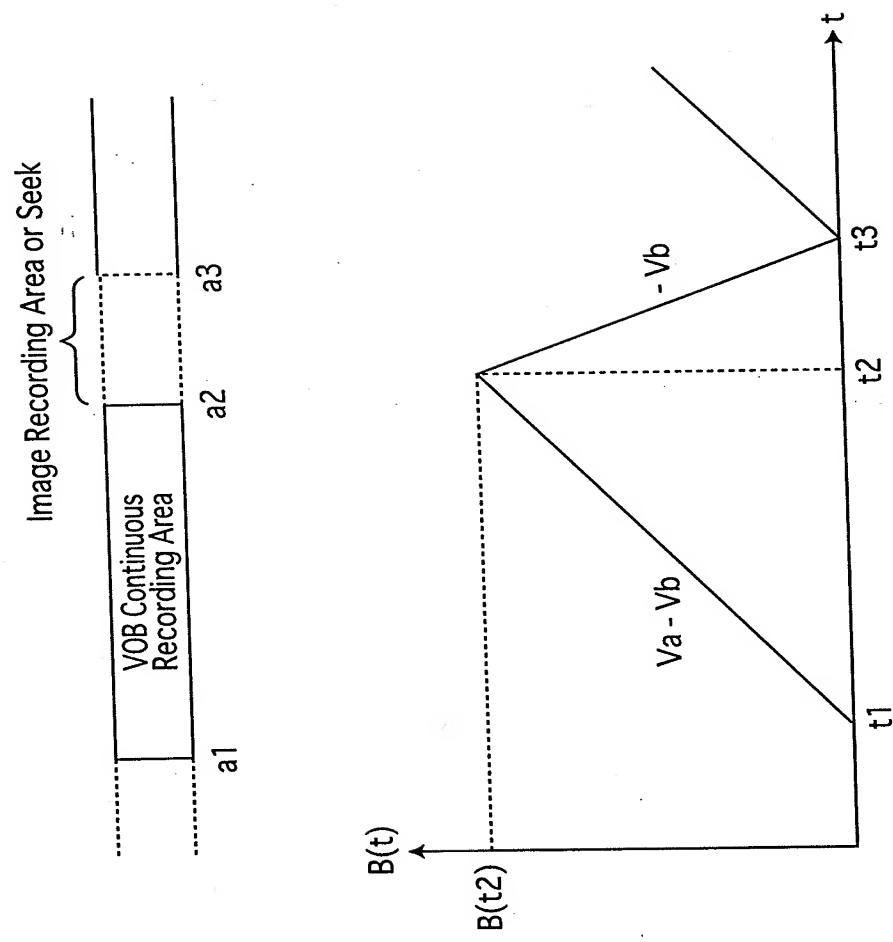


FIG.13

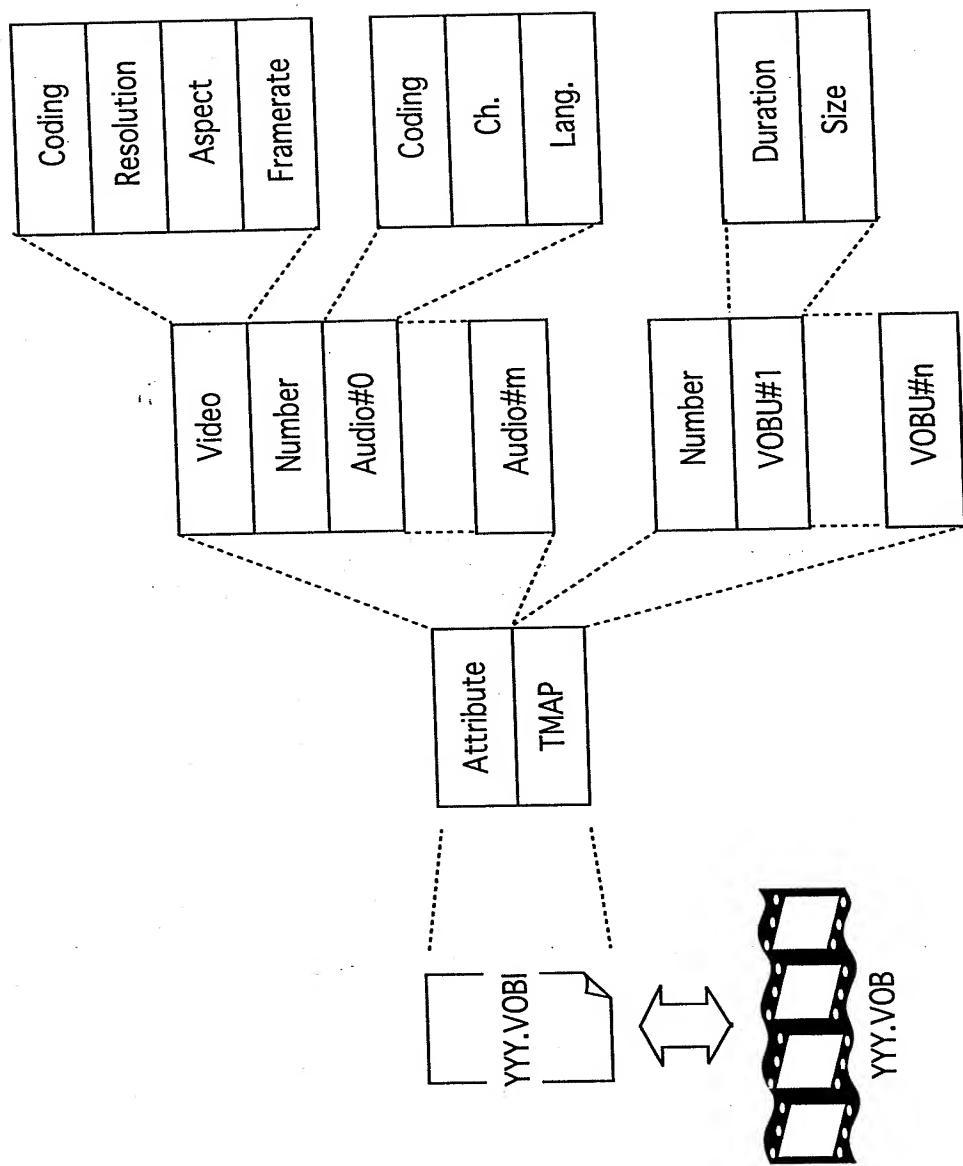


FIG.14

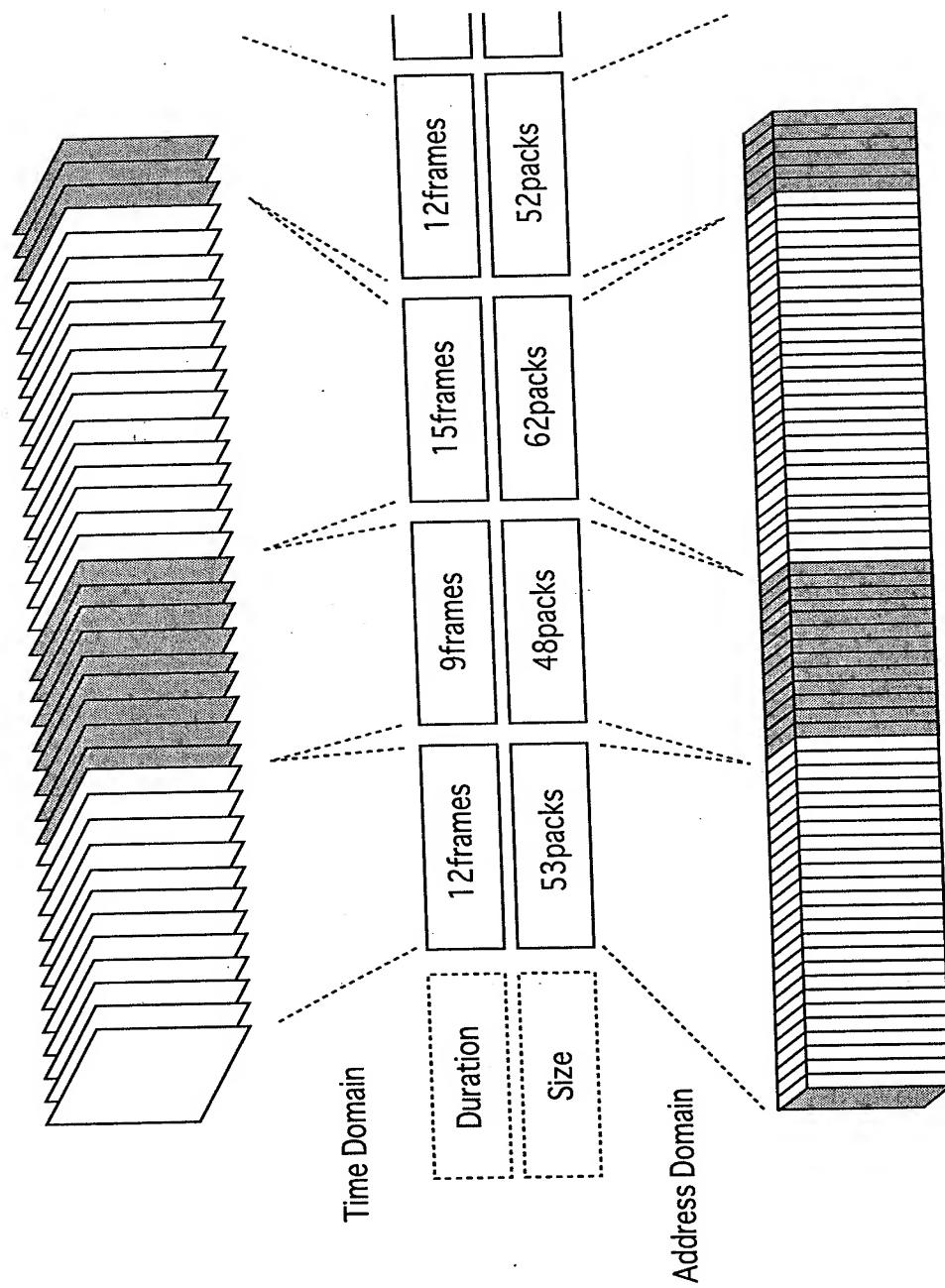


FIG15

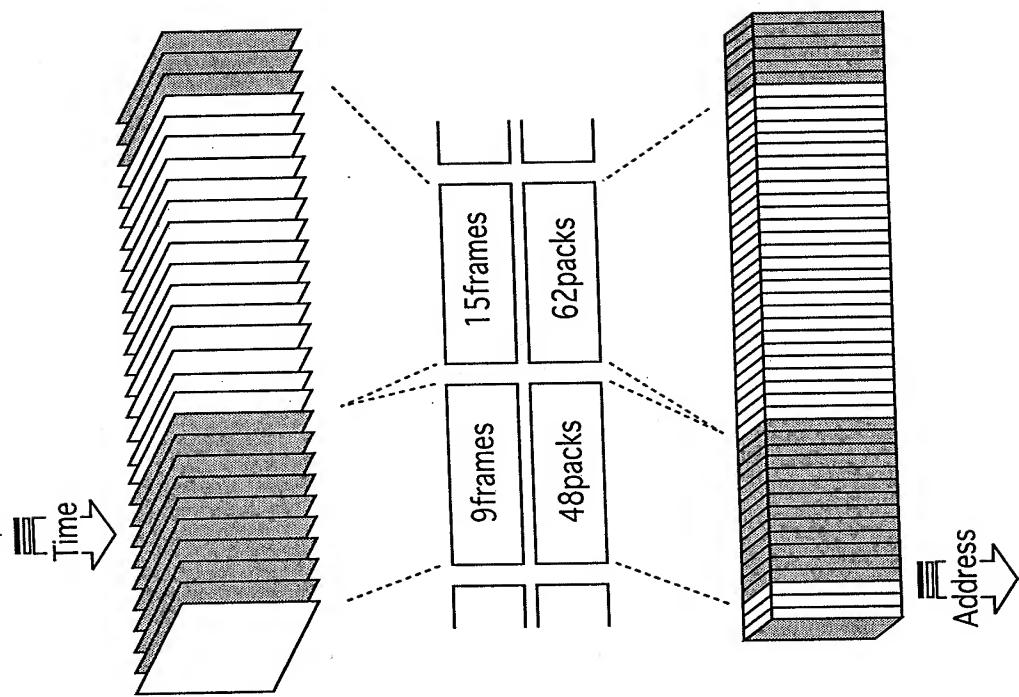


FIG.16

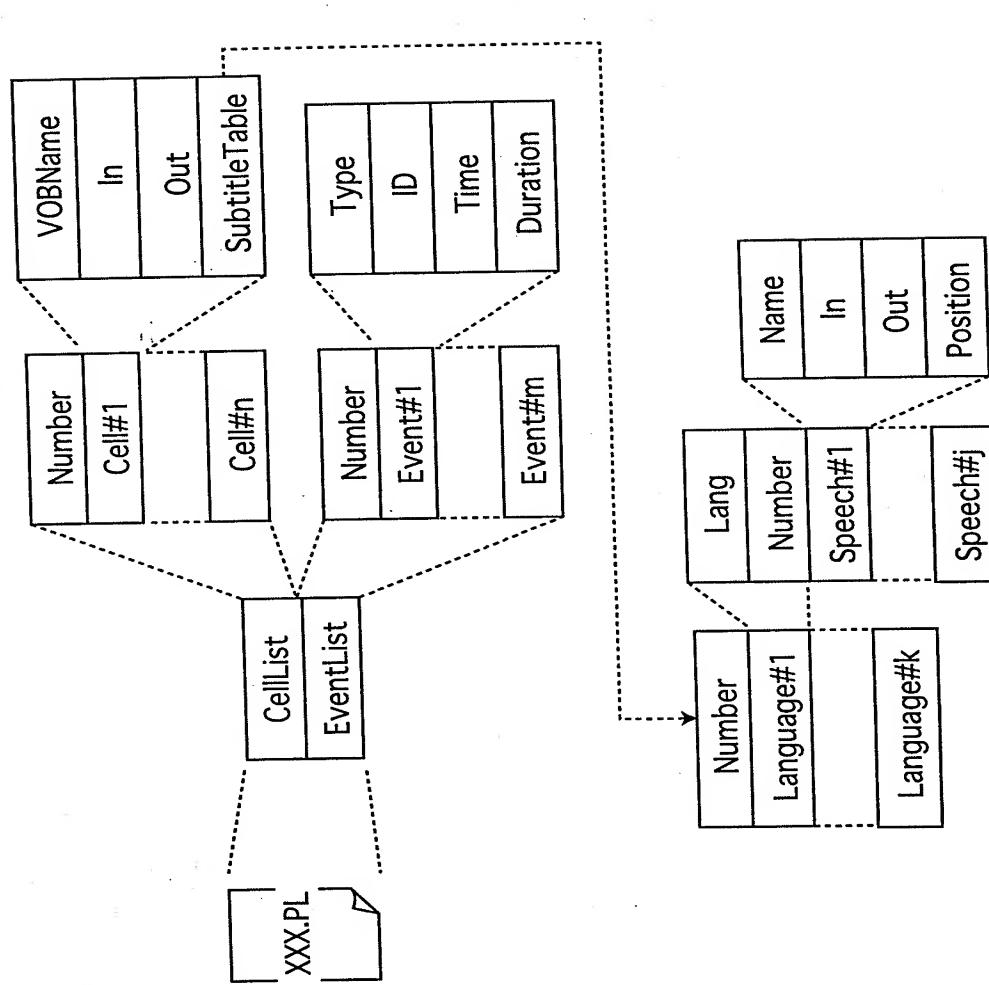


FIG.17

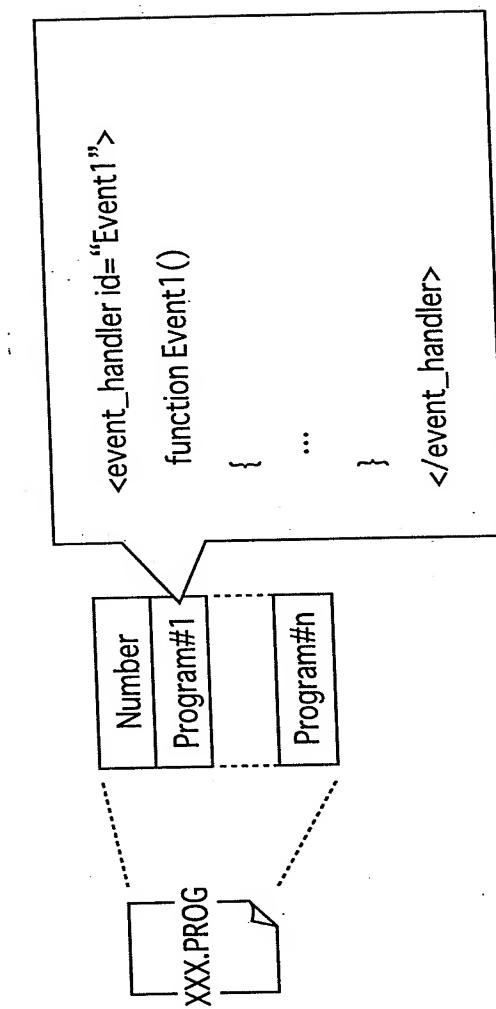


FIG.18

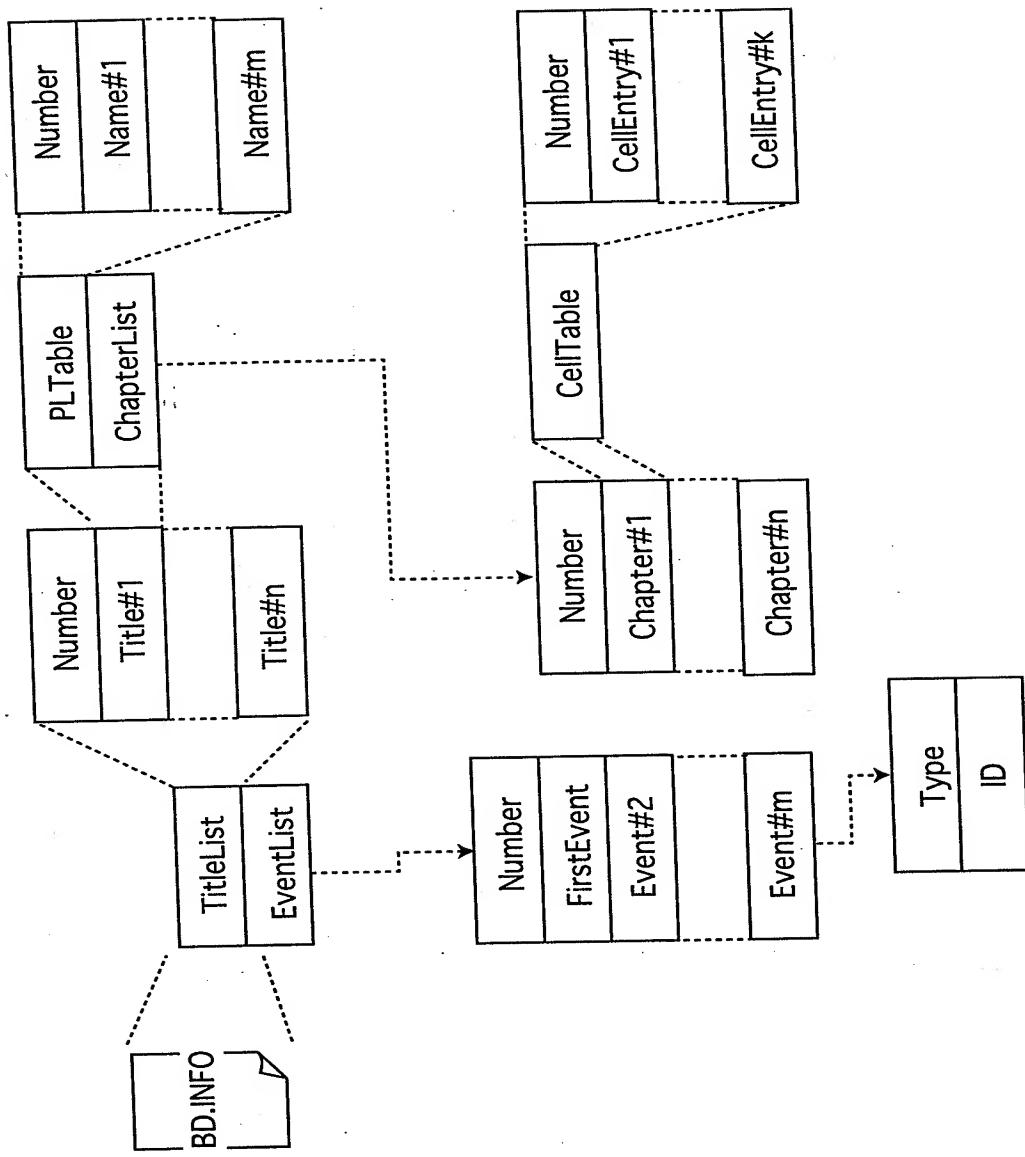


FIG.19

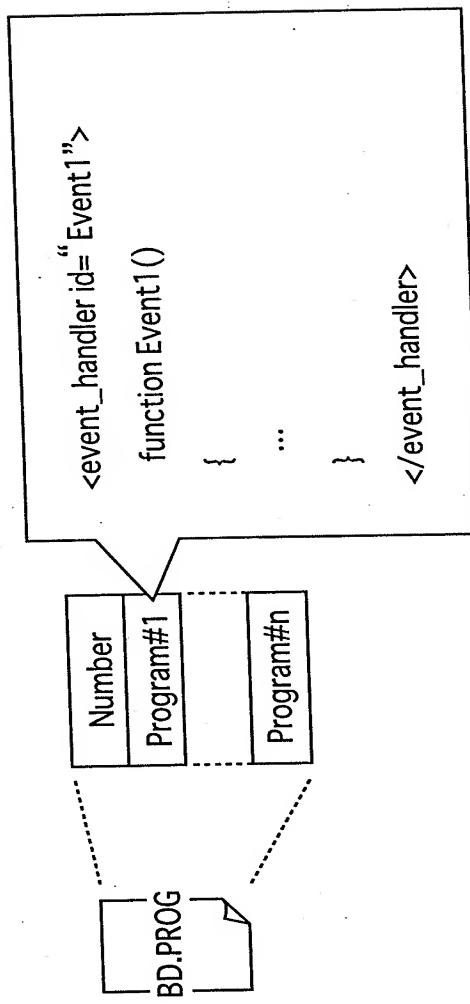


FIG.20

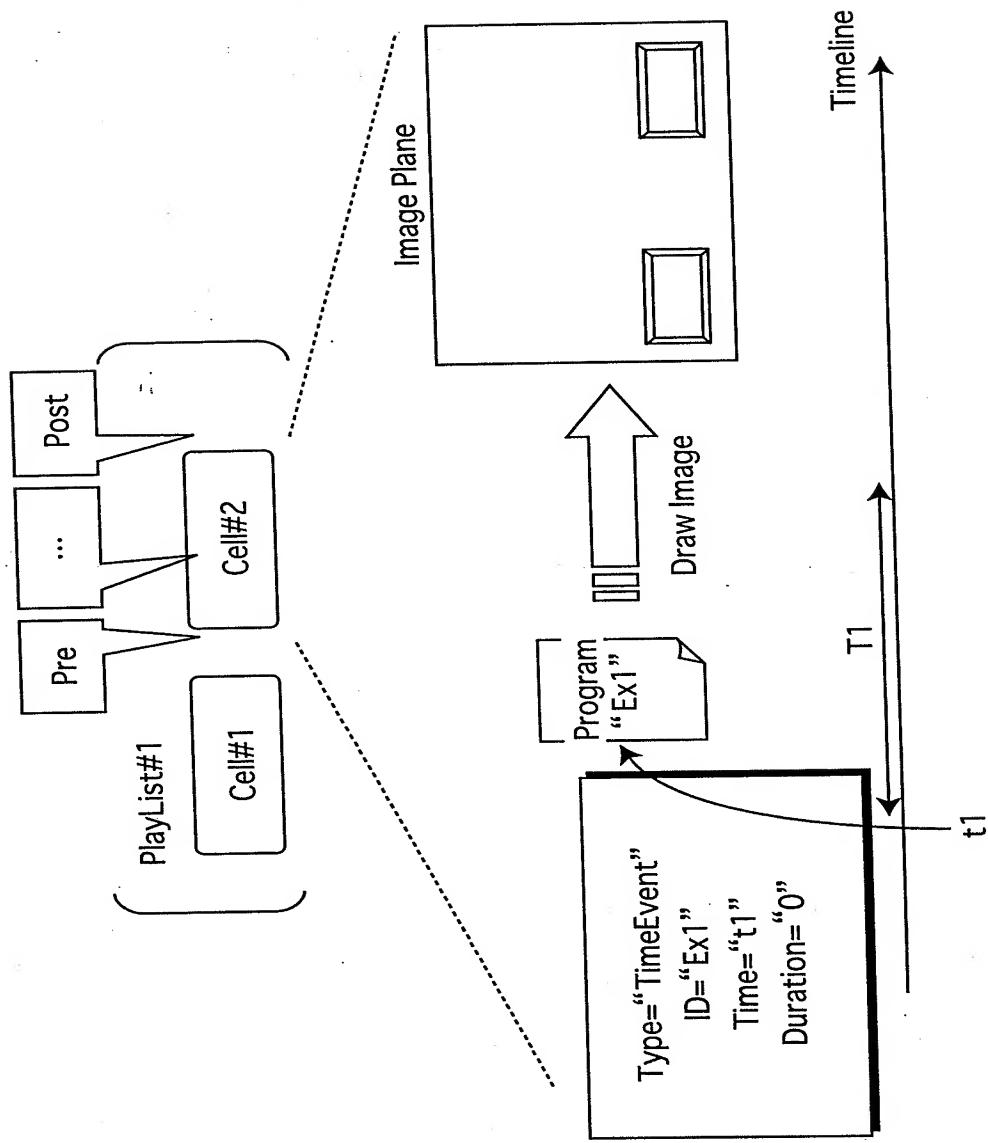


FIG.21

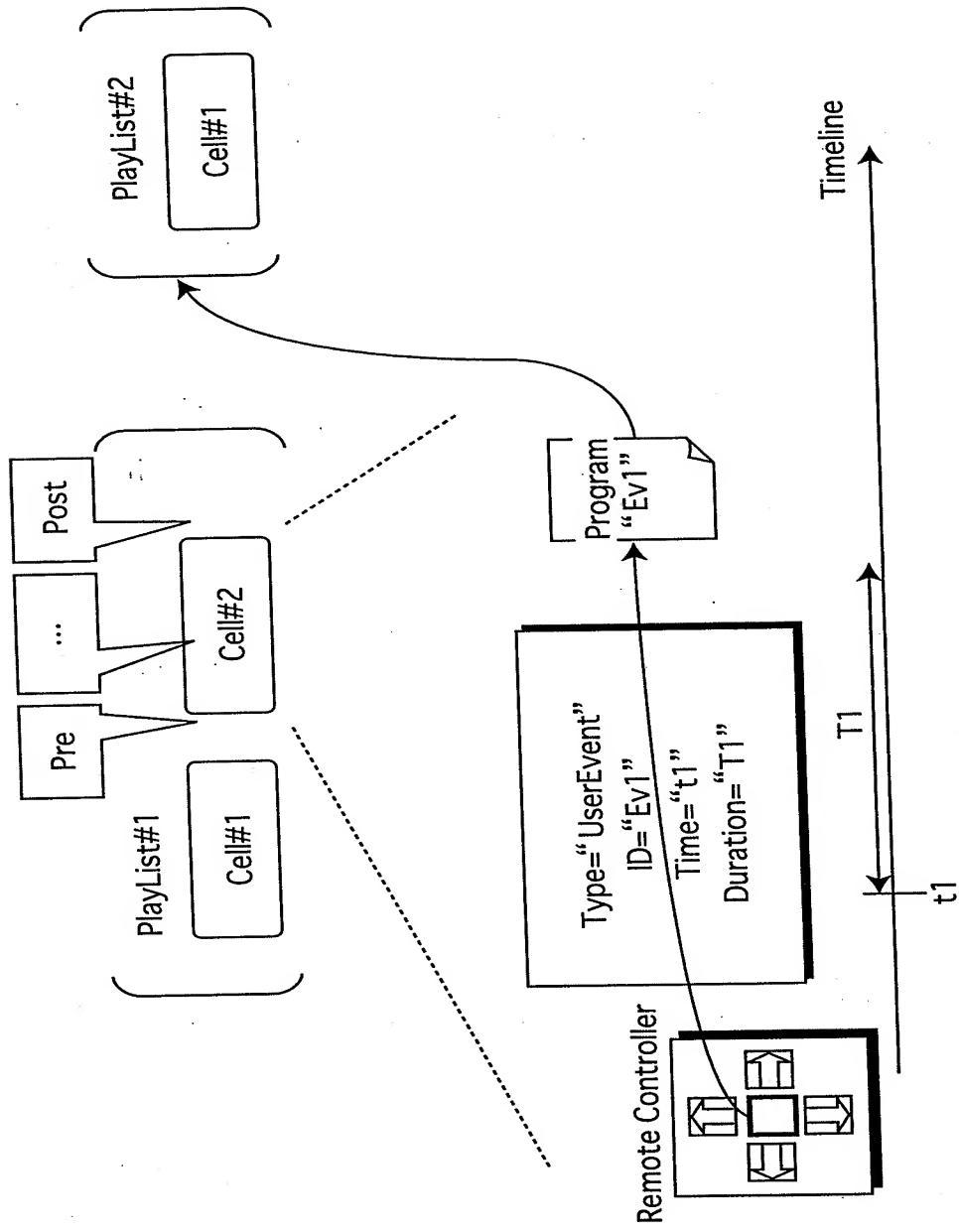


FIG.22

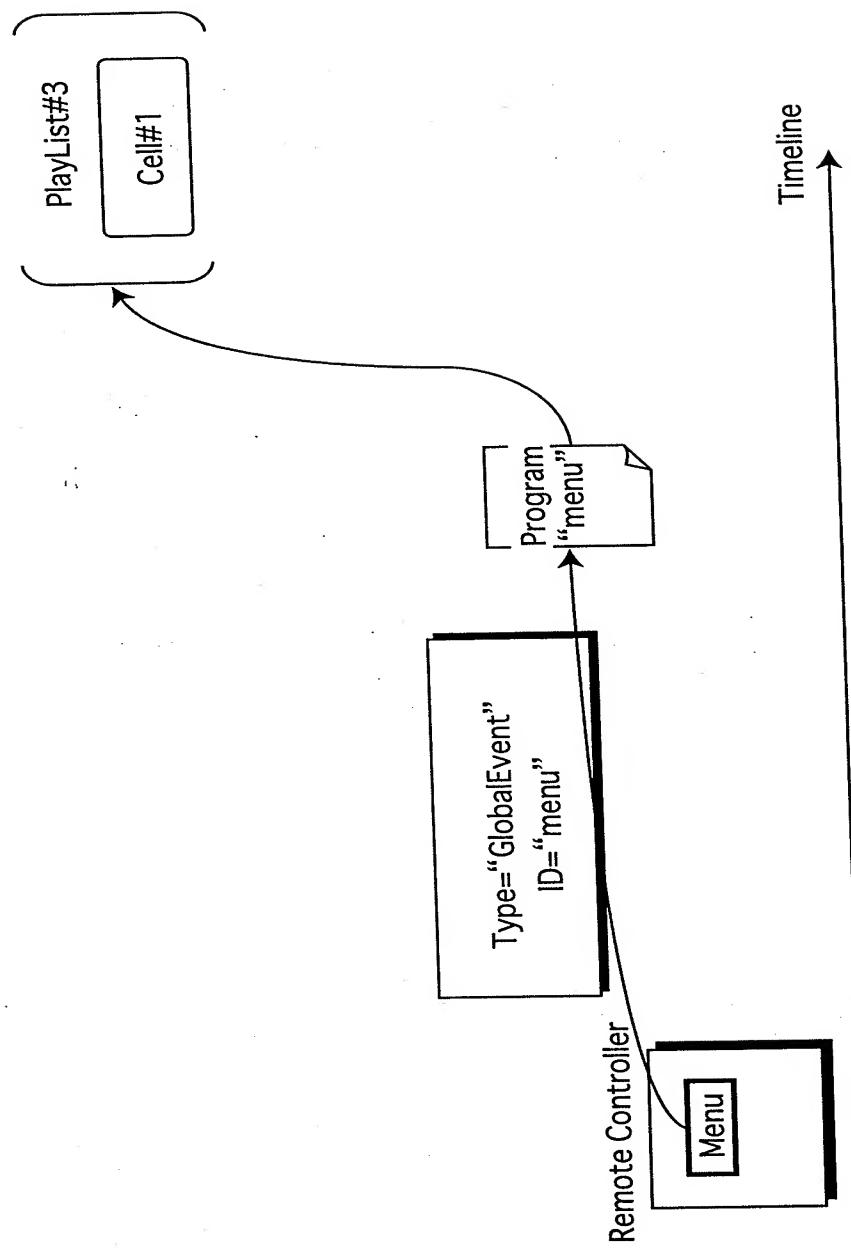


FIG.23

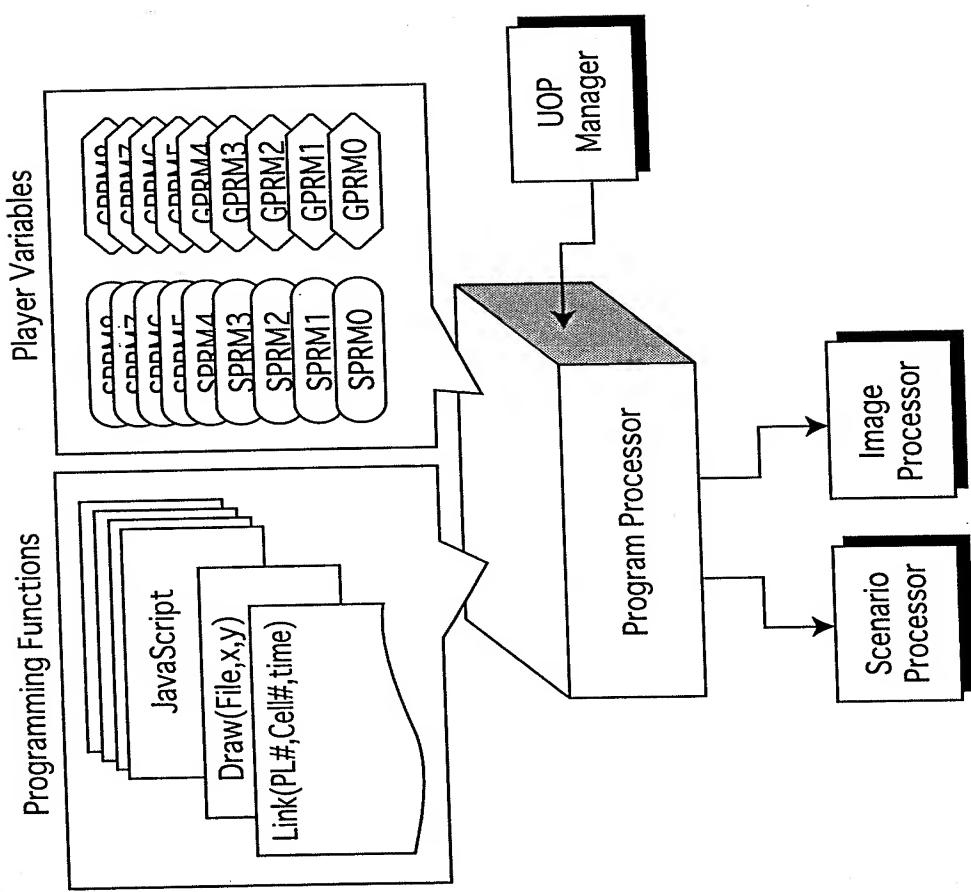


FIG.24  
Player Variables (System Parameters)

0	Language Code	11	Player audio mixing mode for Karaoke	22	reserved
1	Audio stream number	12	Country code for parental management	23	Player status
2	Subtitle stream number	13	Parental level	24	reserved
3	Angle number	14	Player configuration for Video	25	reserved
4	Title number	15	Player configuration for Audio	26	reserved
5	Chapter number	16	Language code for AST	27	reserved
6	Program number	17	Language code ext. for AST	28	reserved
7	Cell number	18	Language code for STST	29	reserved
8	Key name	19	Language coded ext. for STST	30	reserved
9	Navigation timer	20	Player region code	31	reserved
10	Current playback time	21	reserved	32	reserved

FIG.25

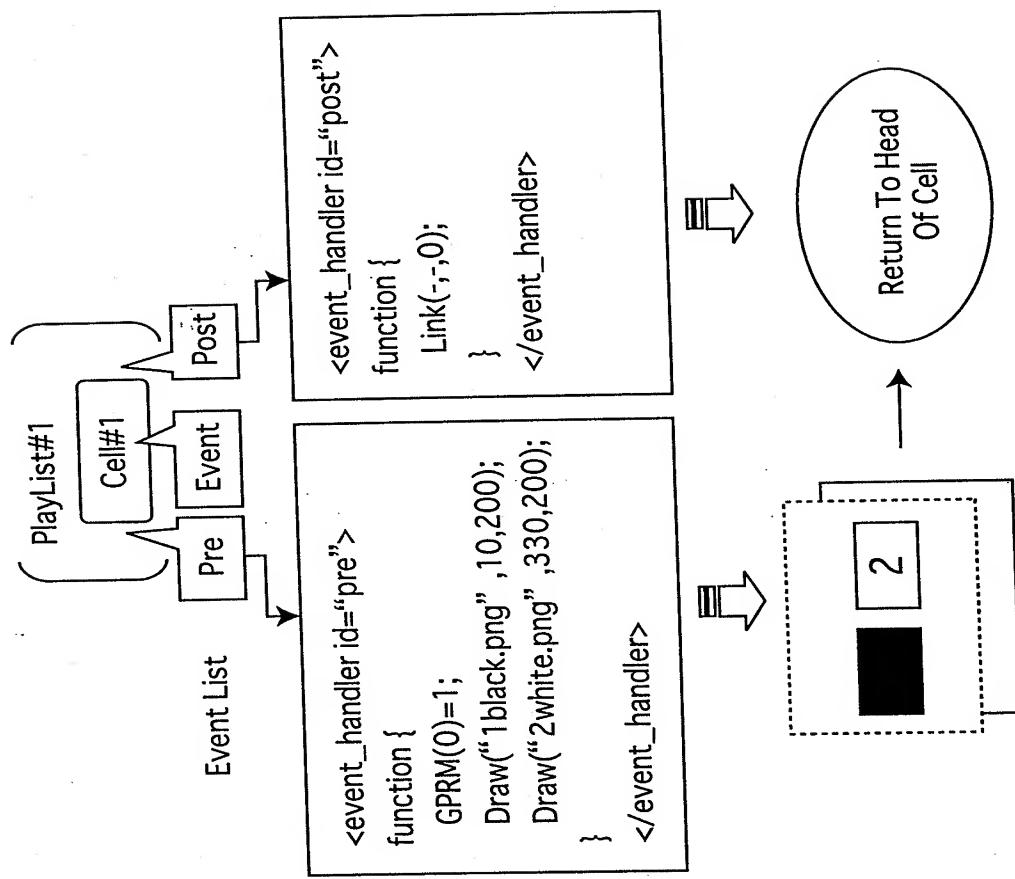


FIG.26

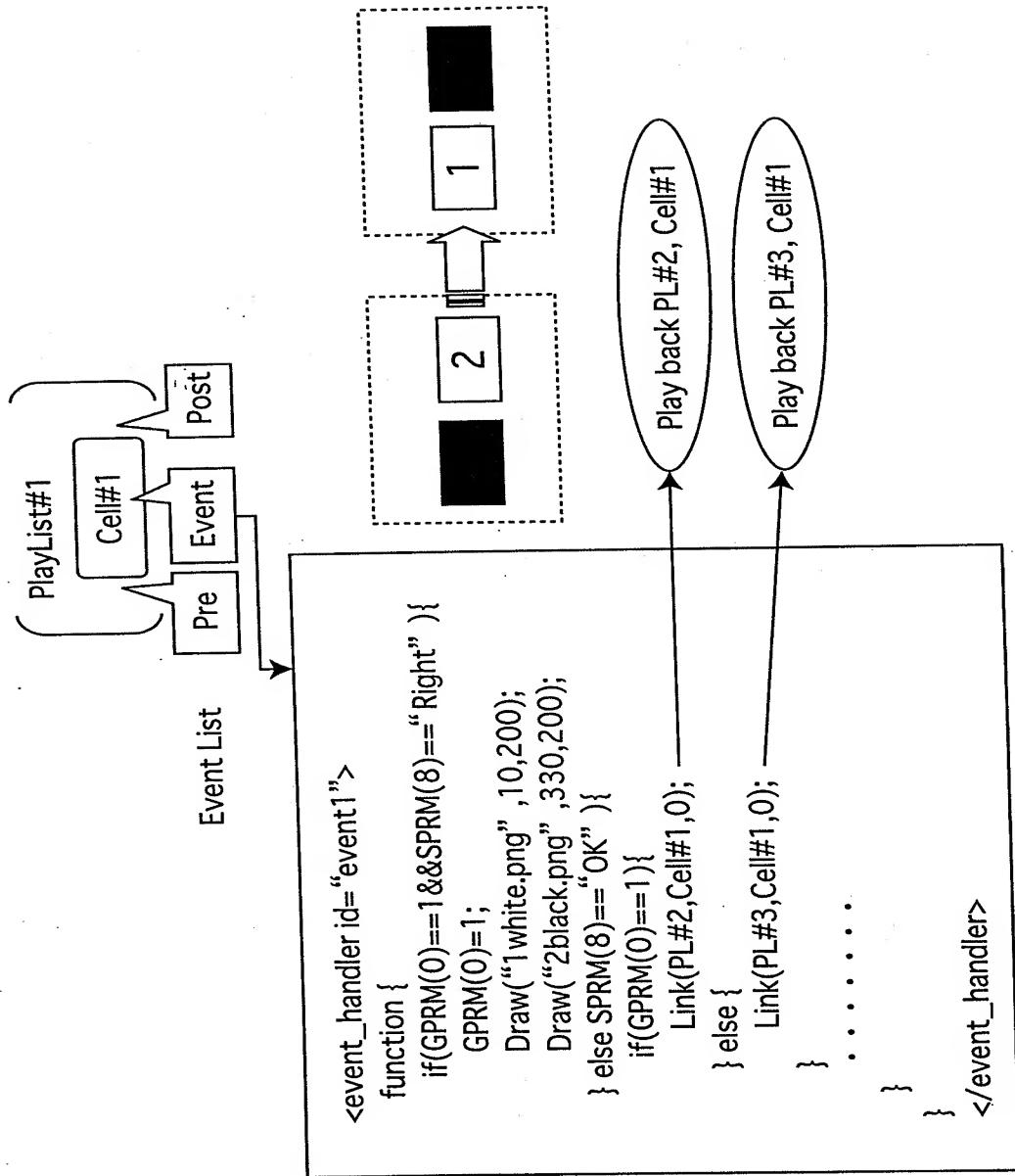


FIG.27

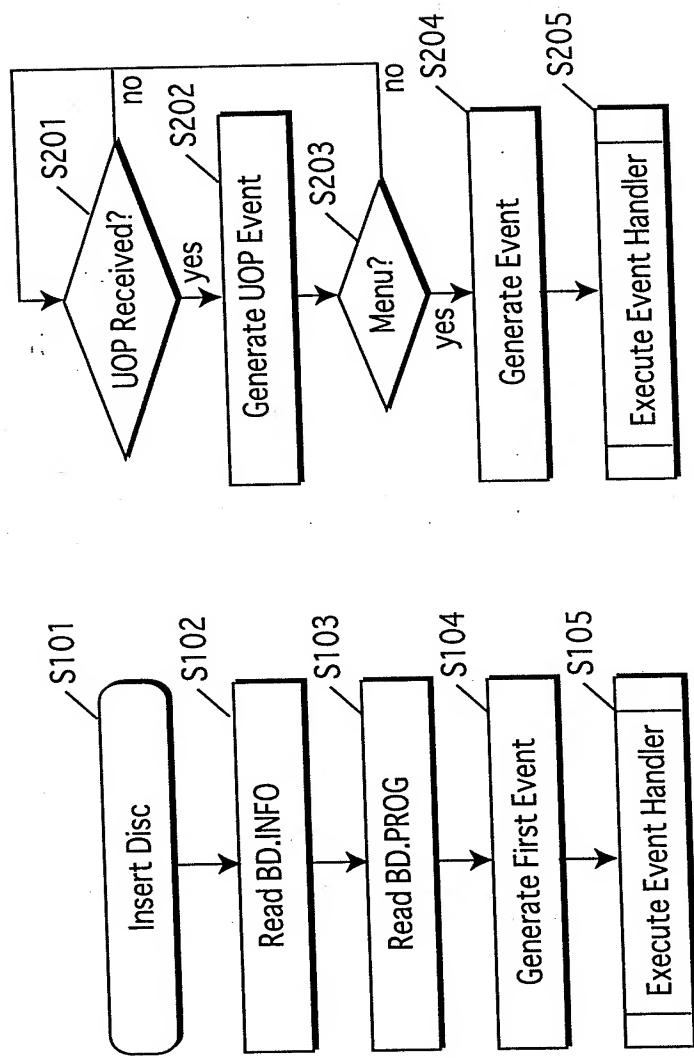


FIG.28

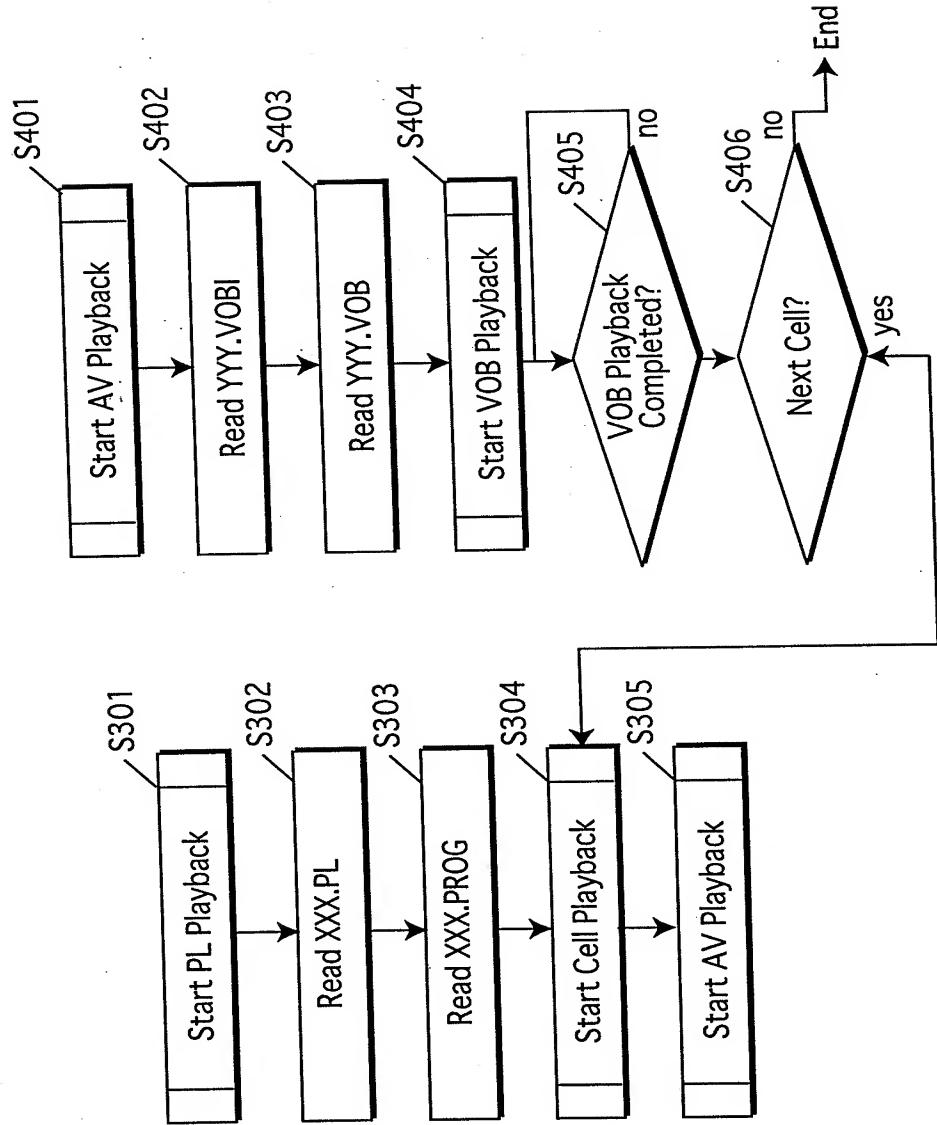


FIG.29

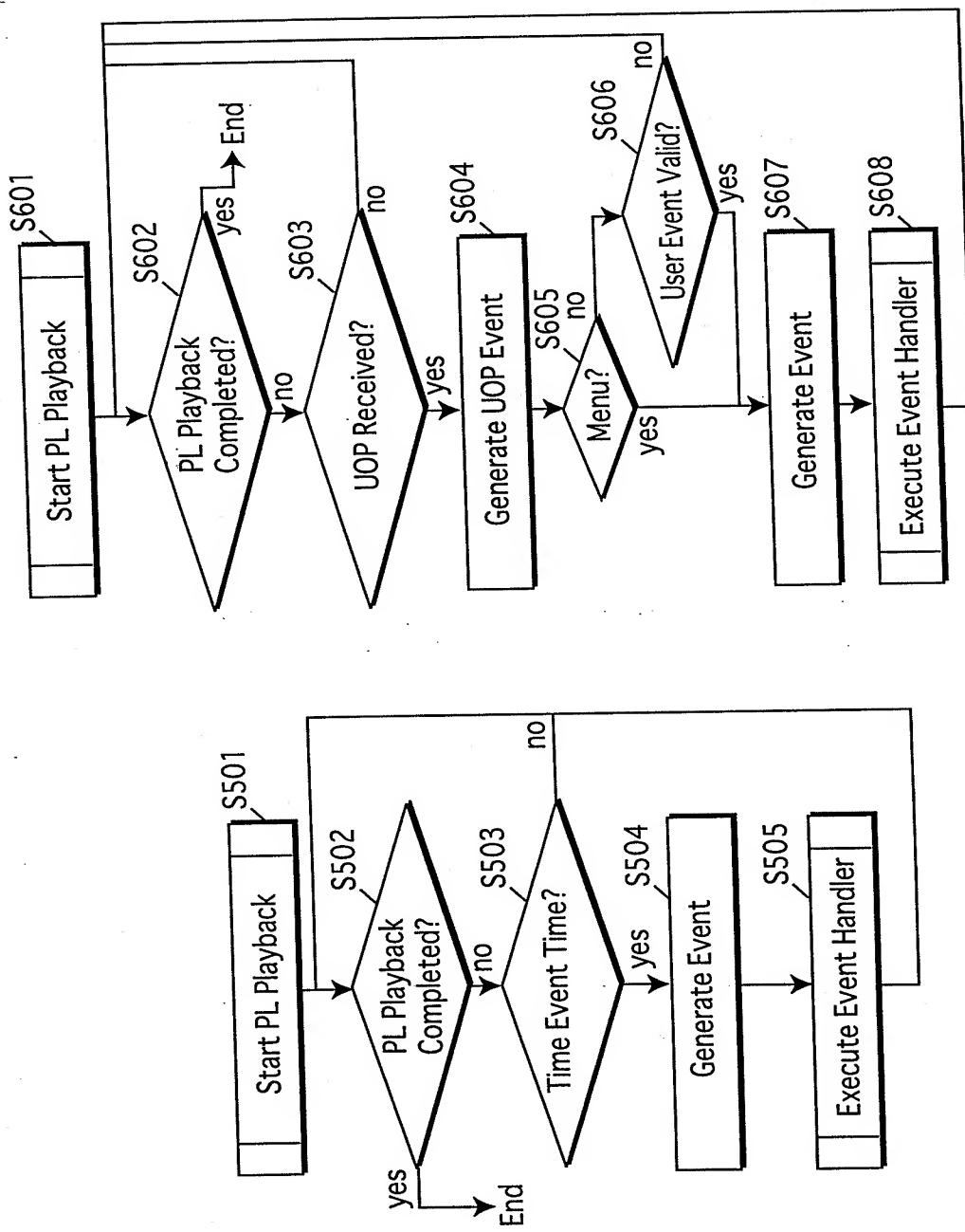


FIG.30

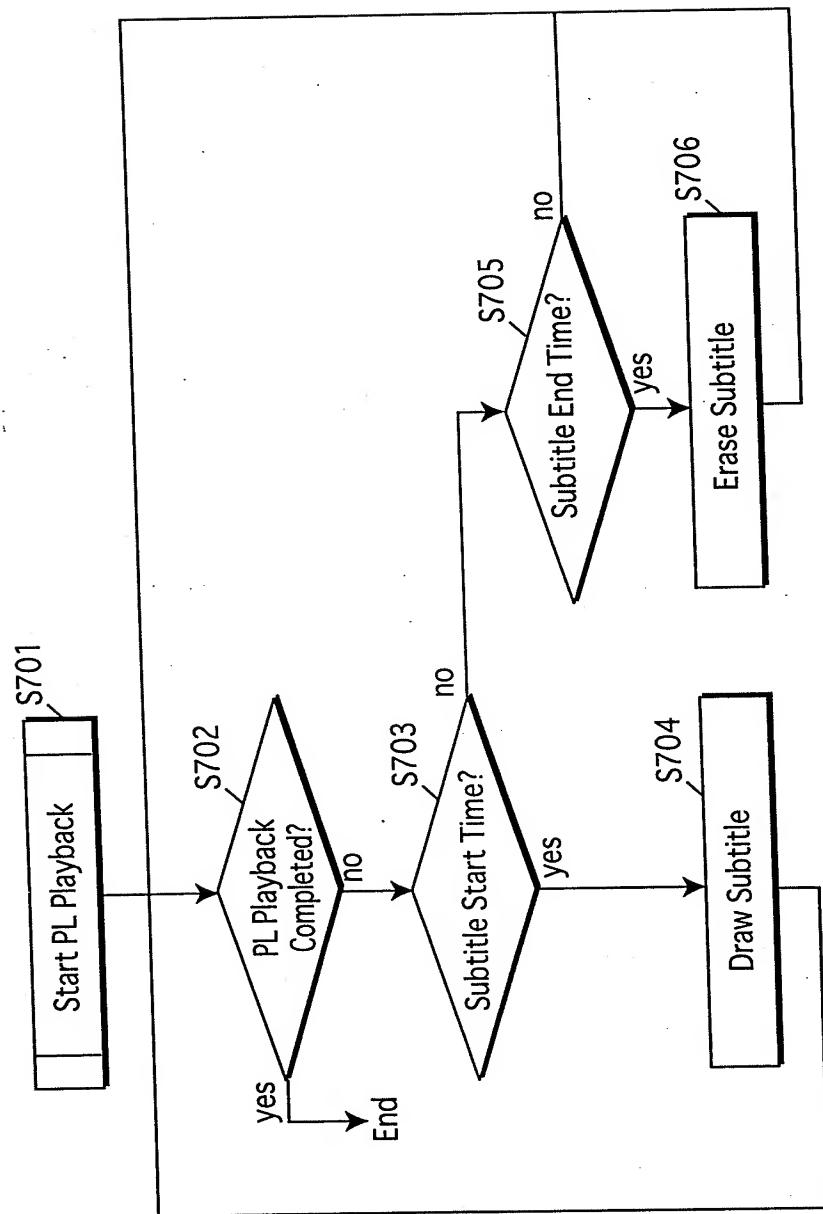


FIG.31

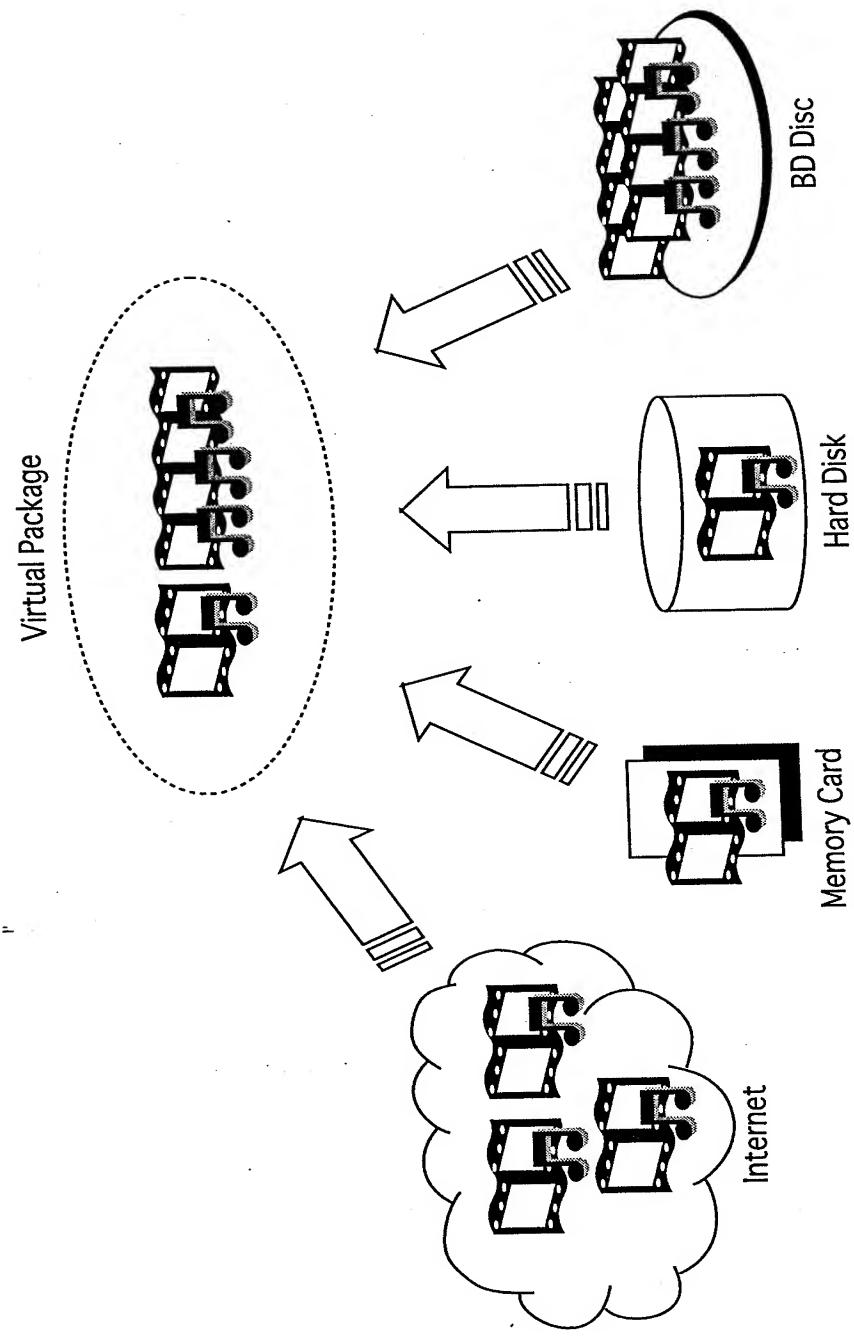


FIG.32

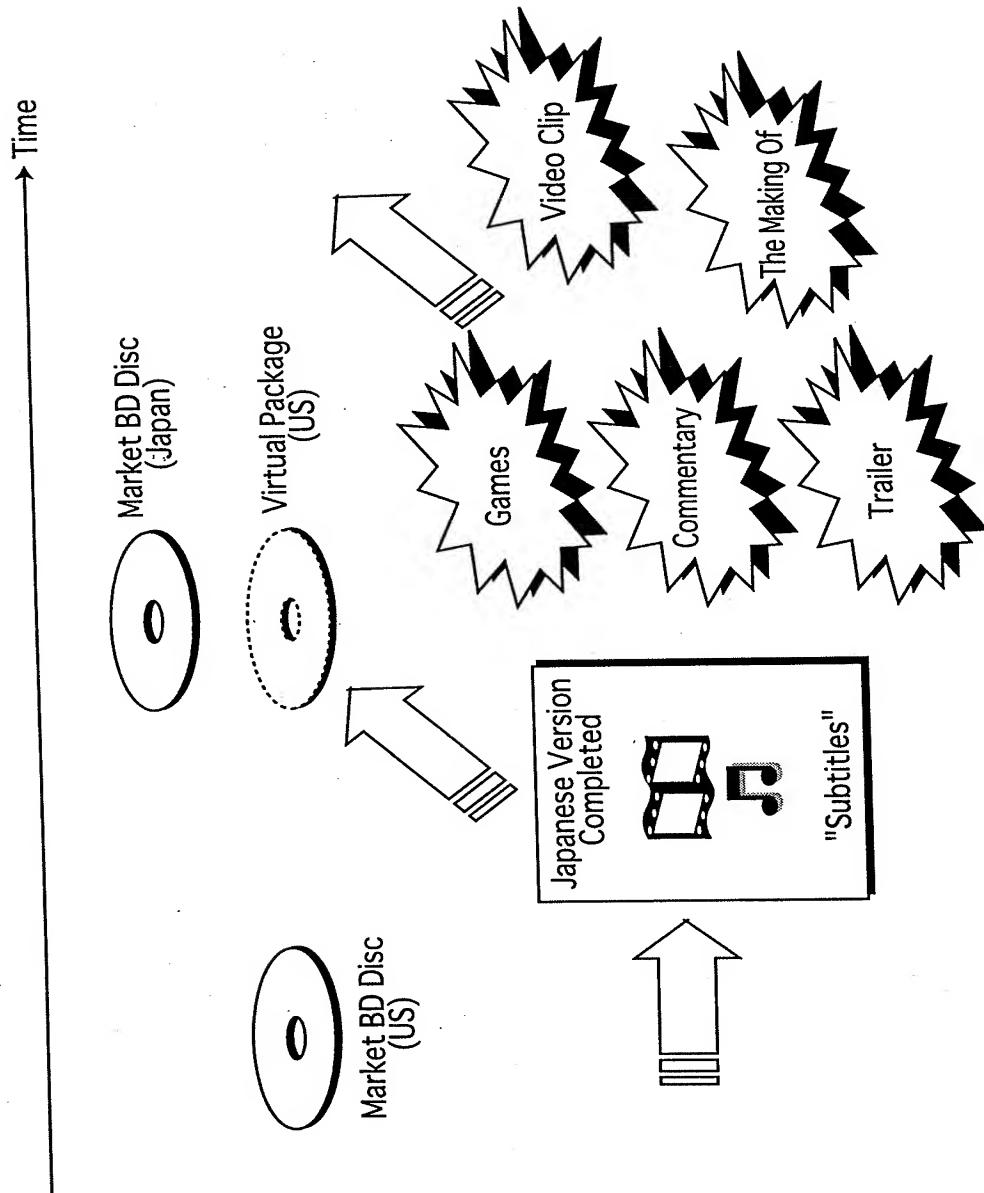


FIG.33

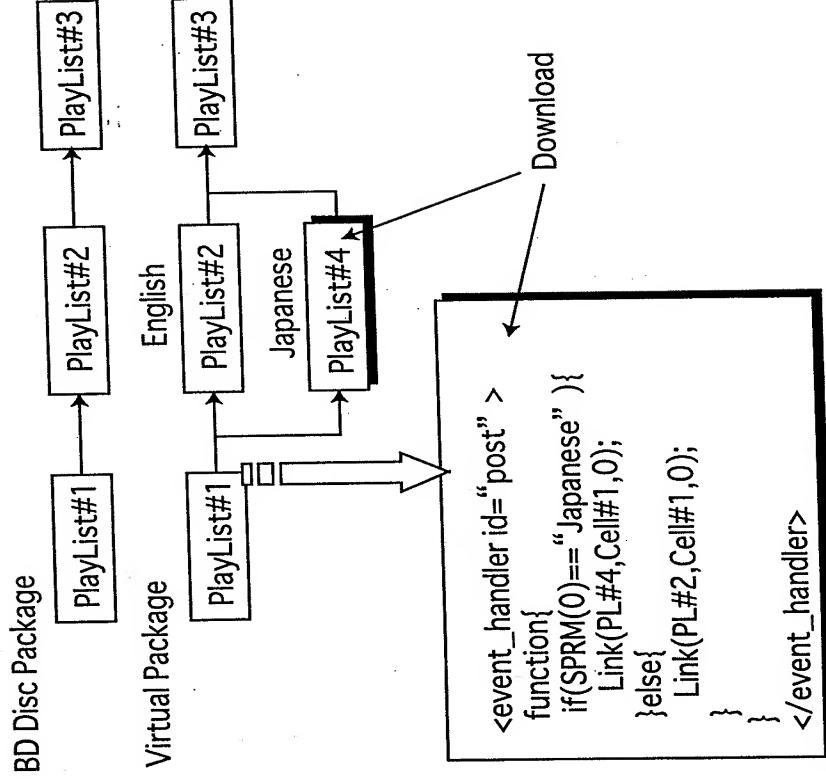


FIG.34

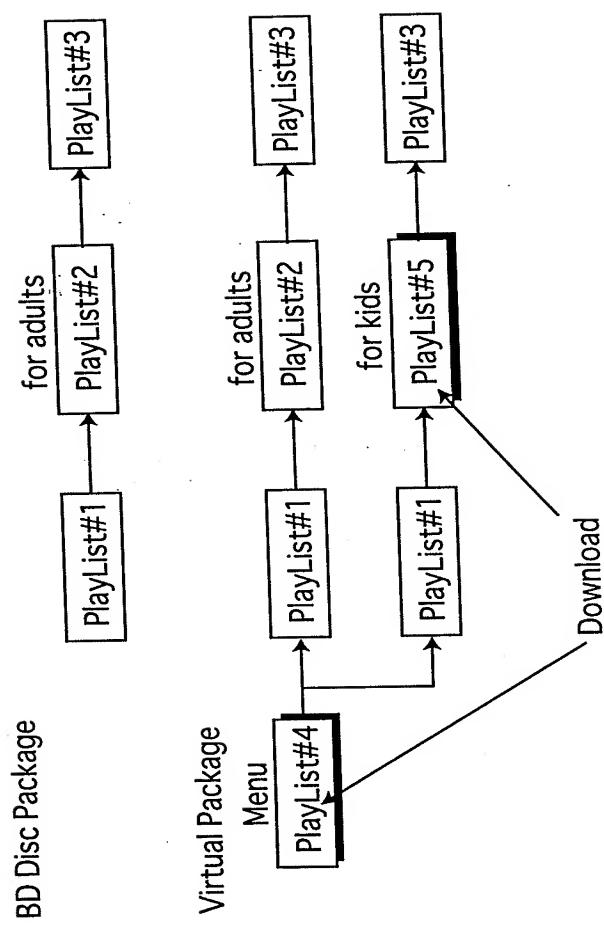


FIG.35

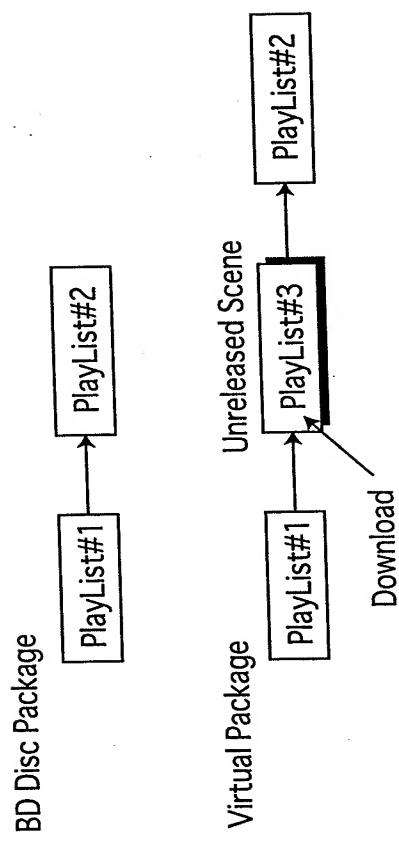


FIG.36

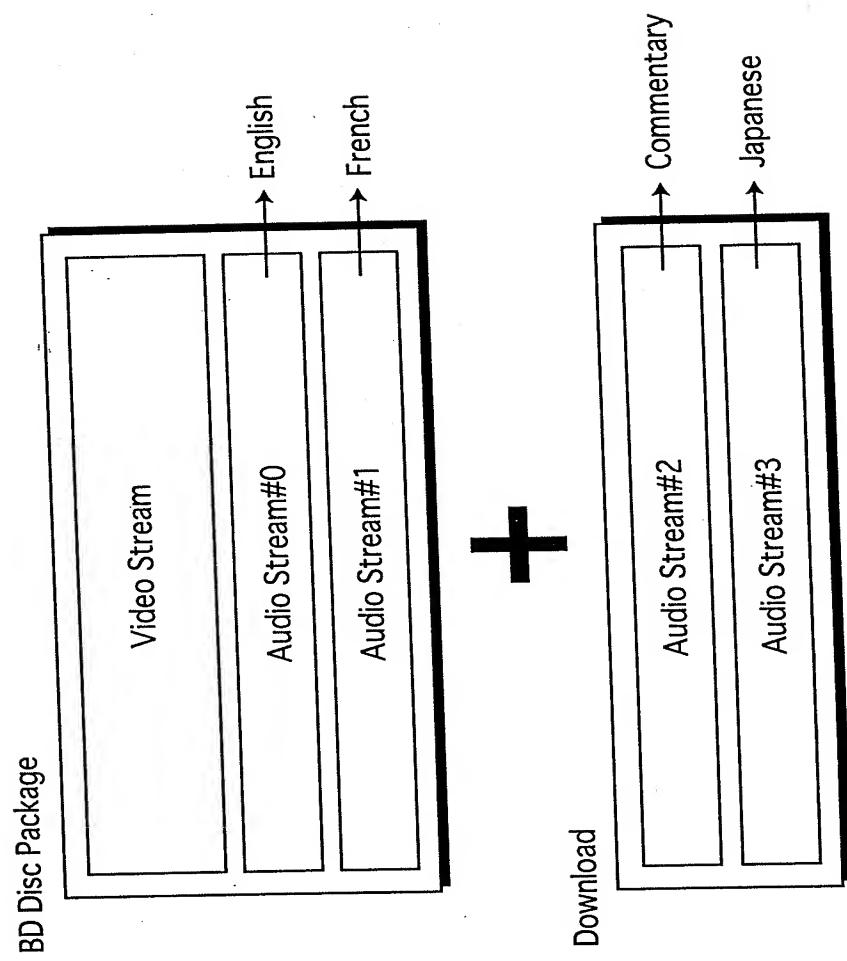


FIG.37

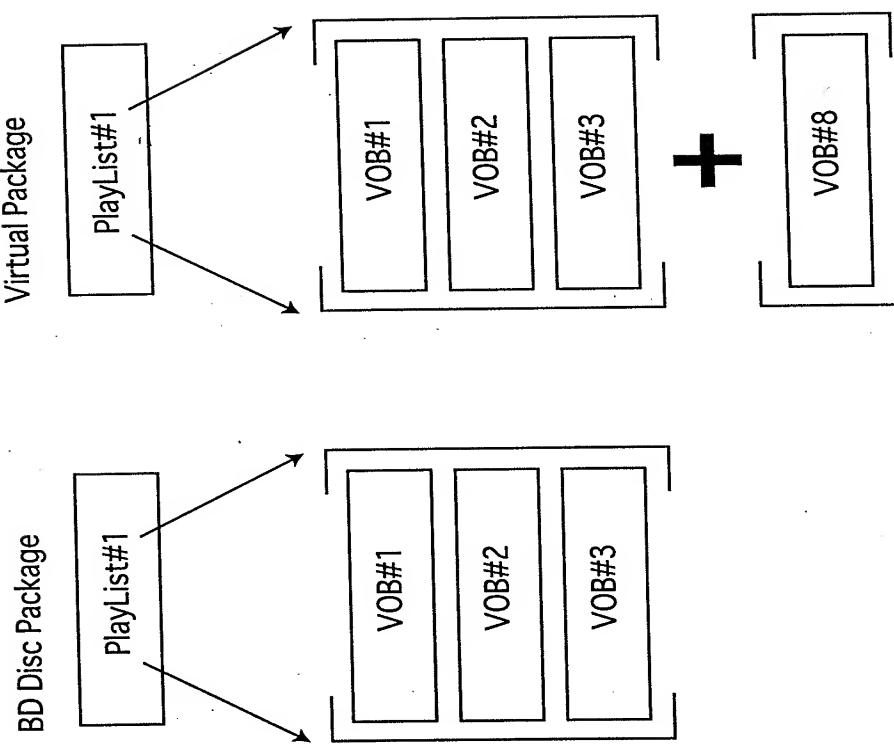


FIG.38

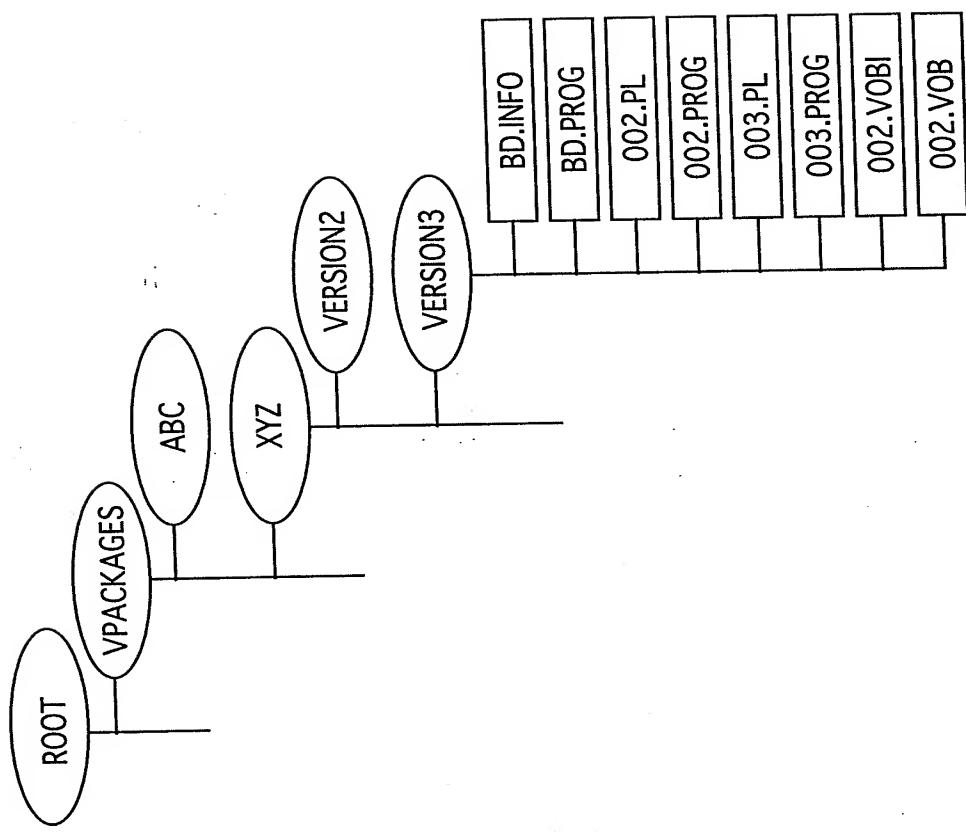


FIG.39

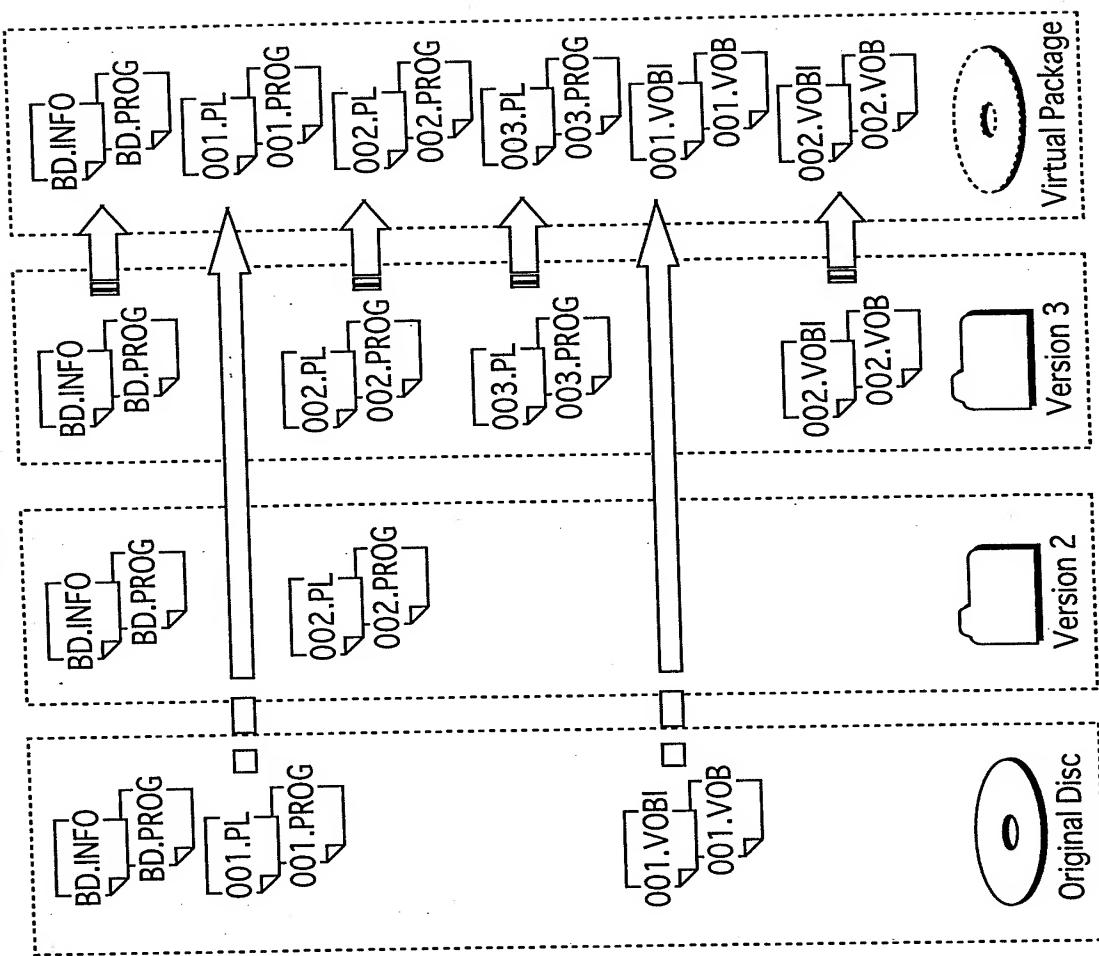


FIG. 40

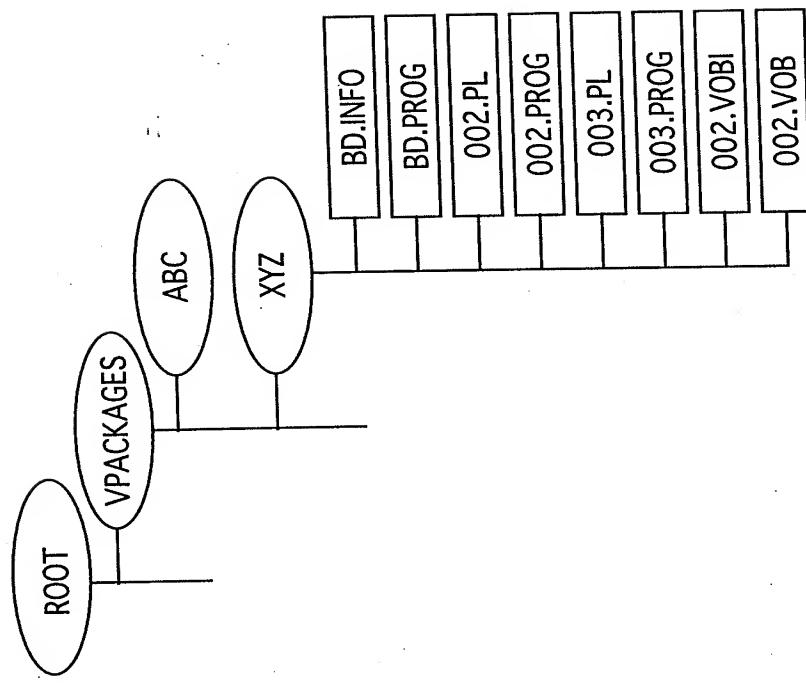


FIG.41

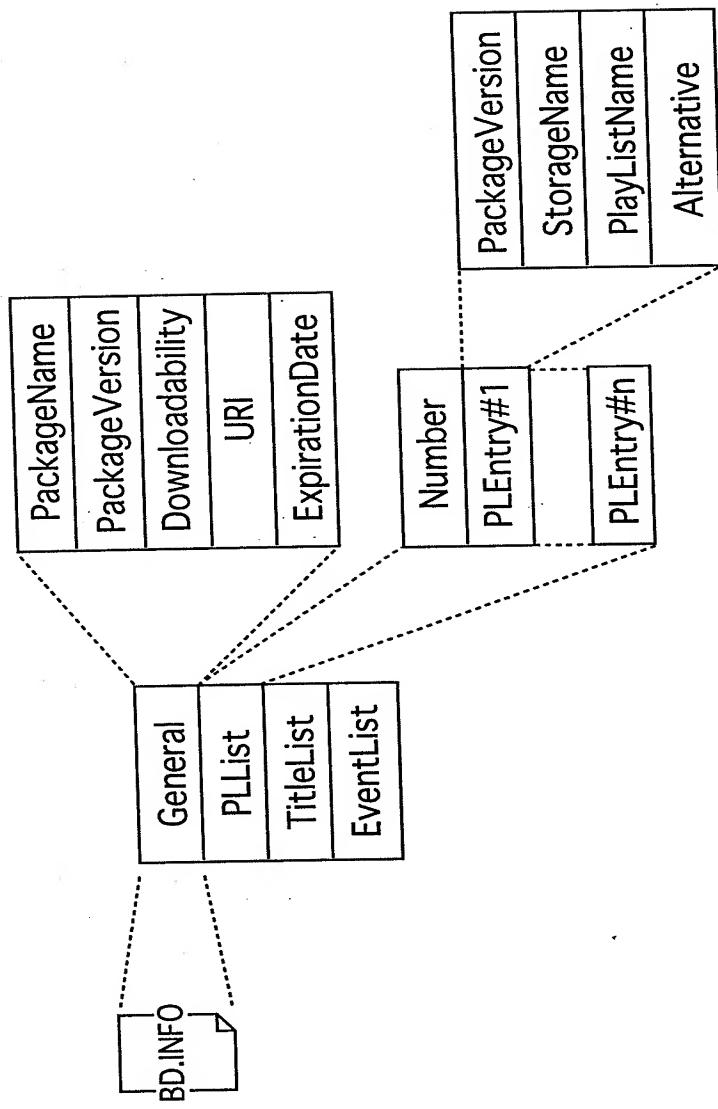


FIG.42

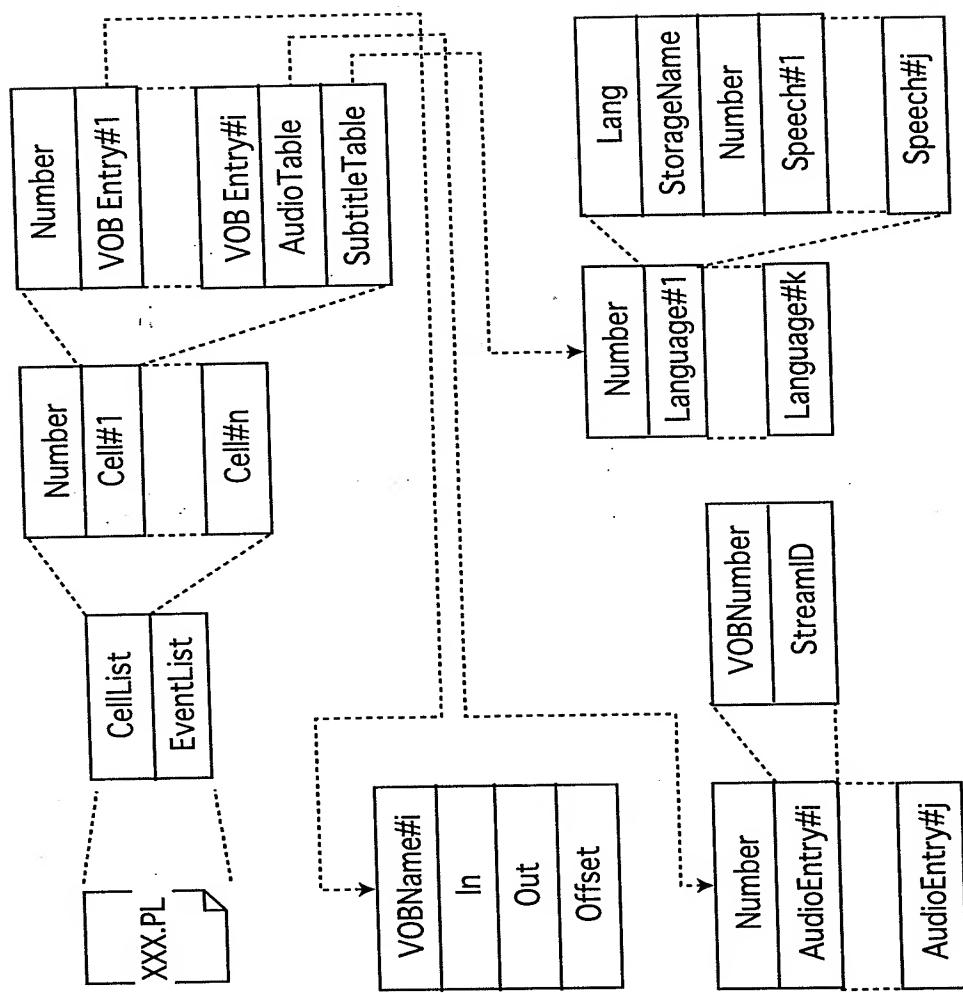


FIG.43

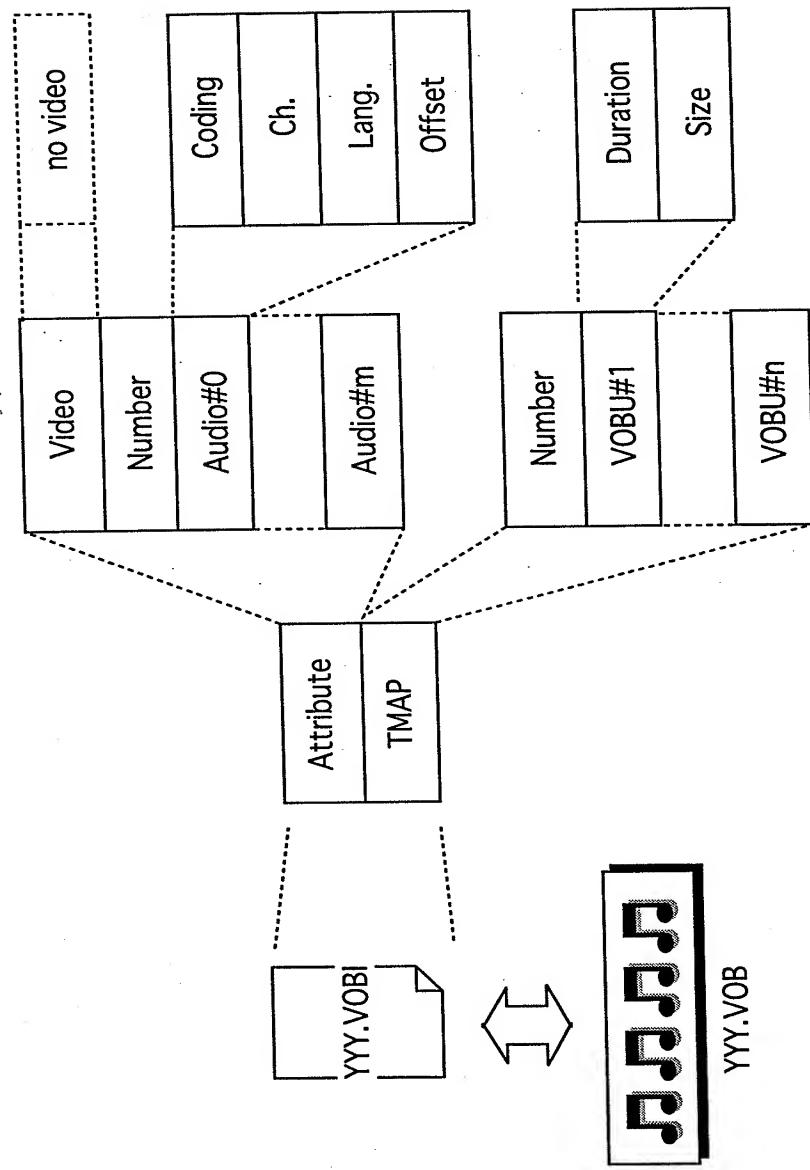


FIG.44

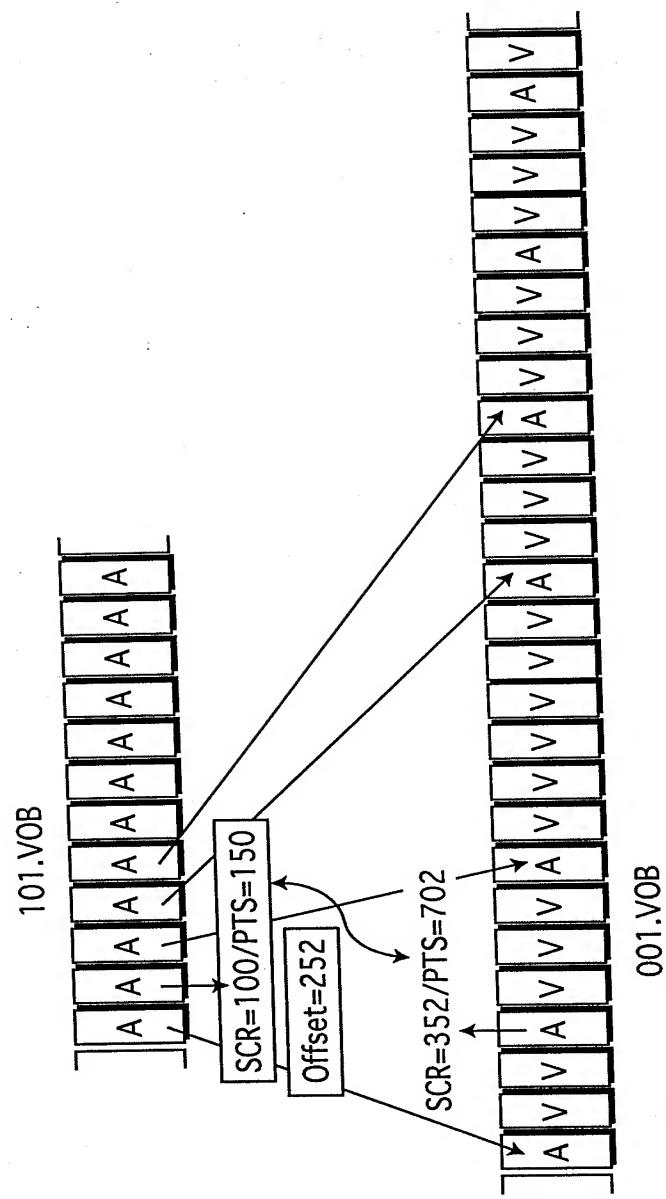


FIG.45

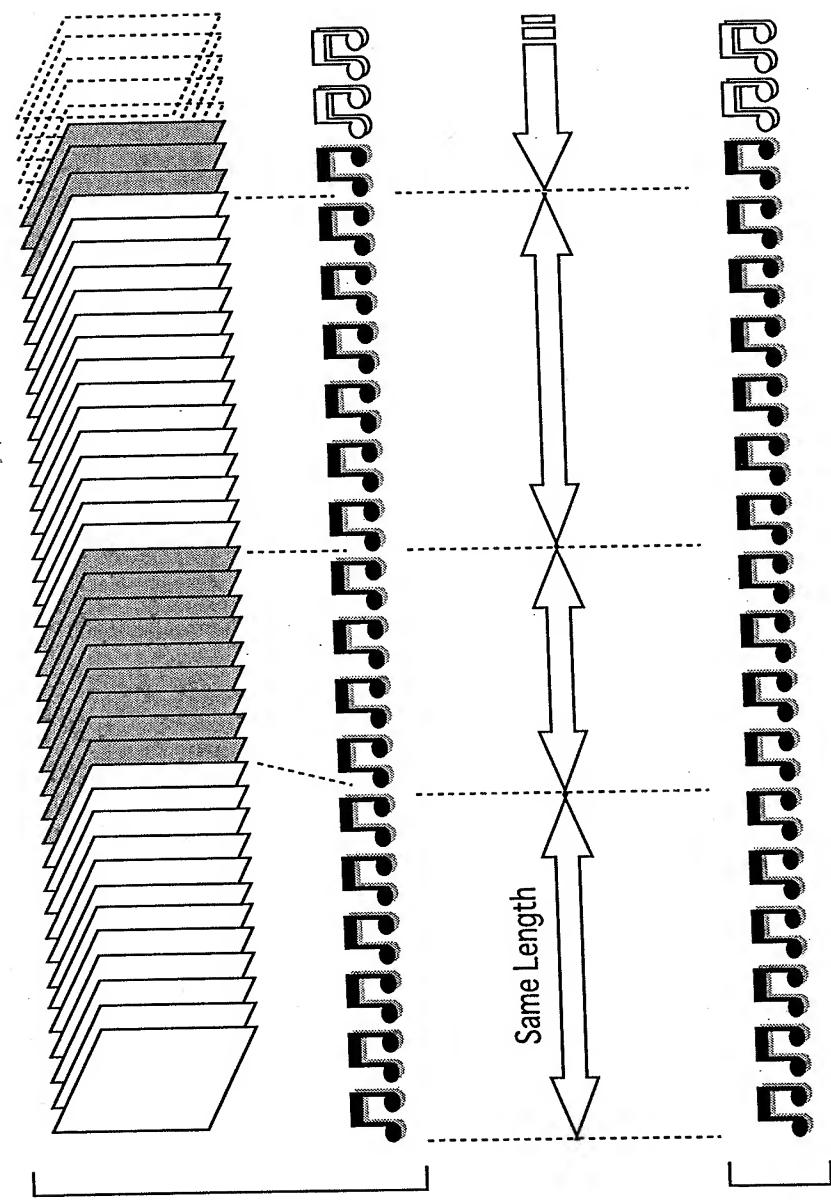


FIG.46

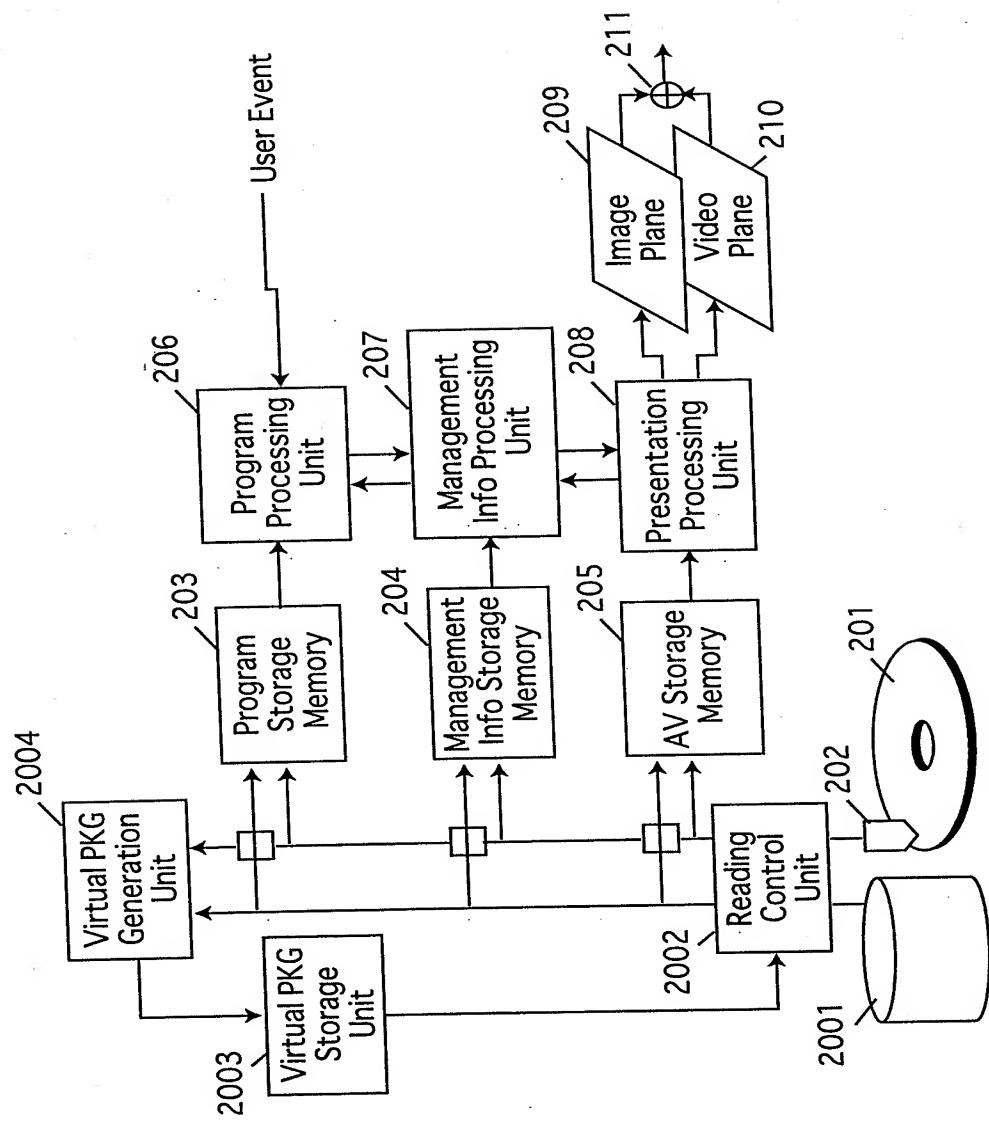


FIG.47

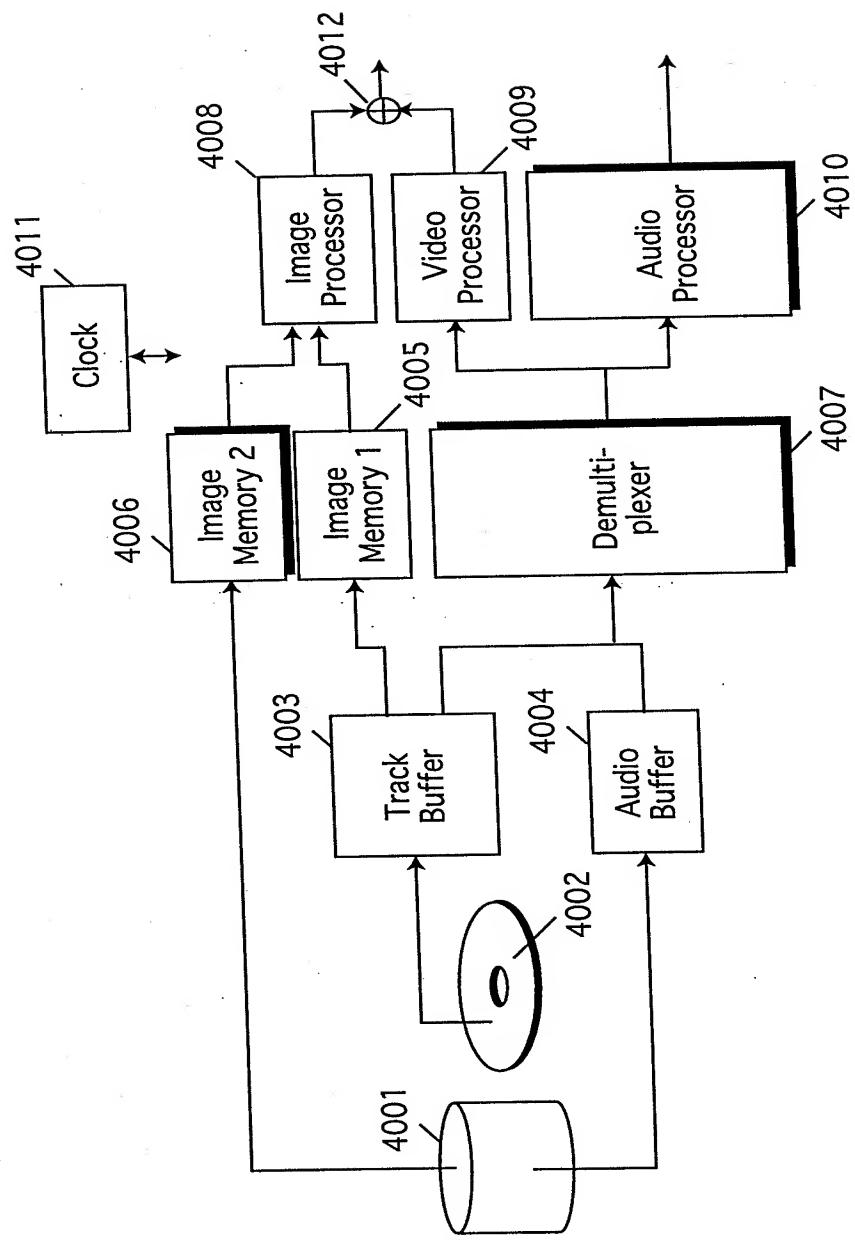


FIG.48

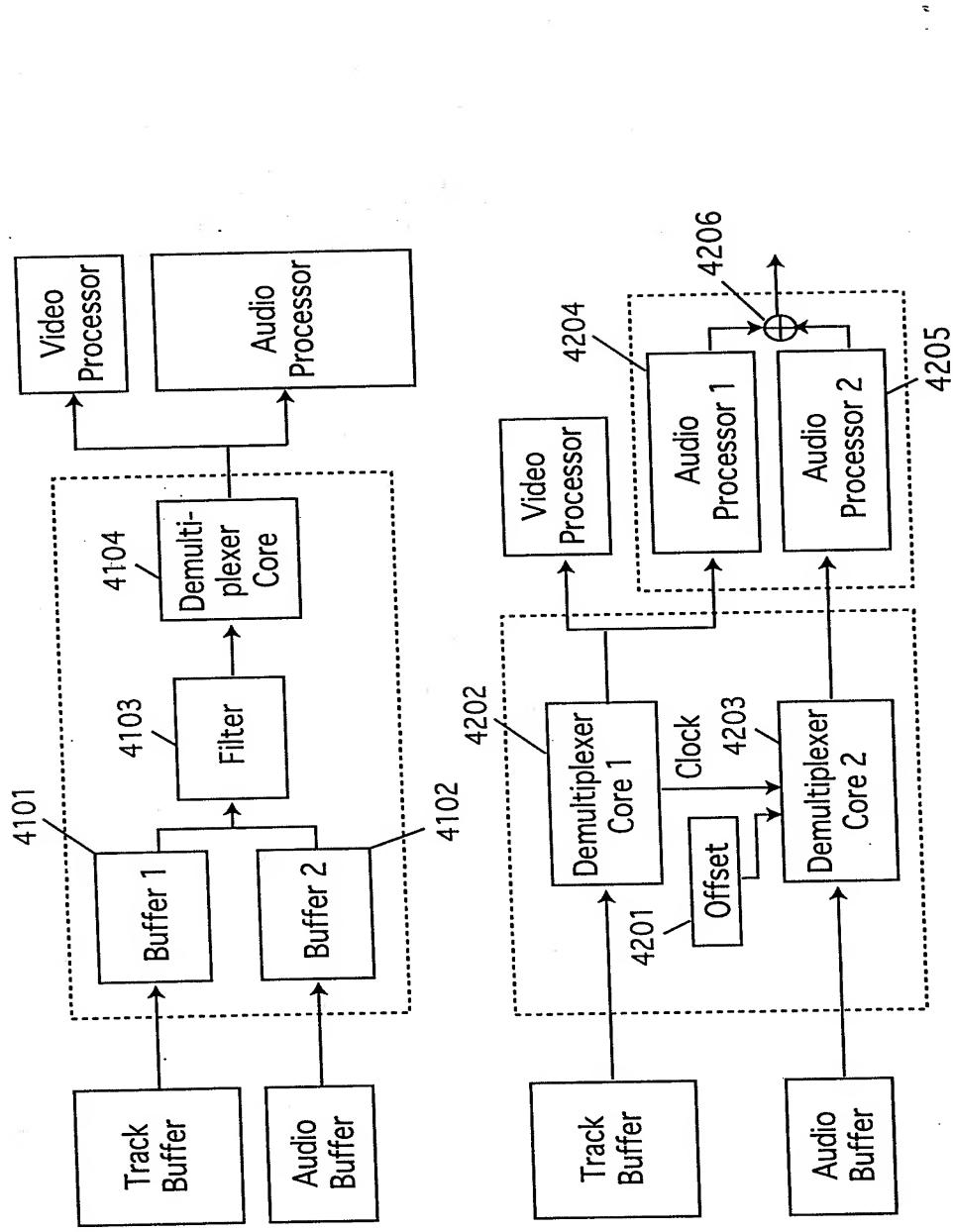


FIG.4.9

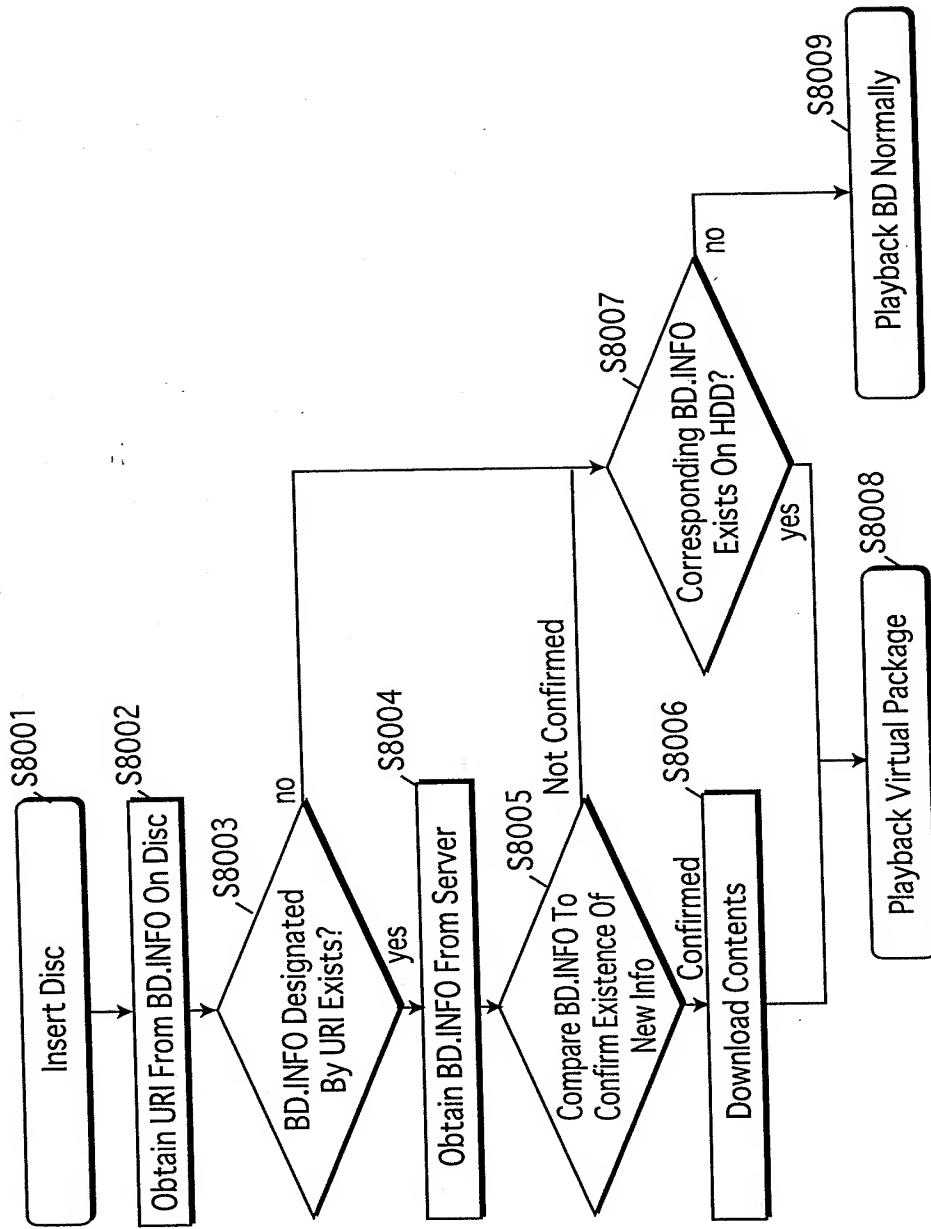


FIG.50

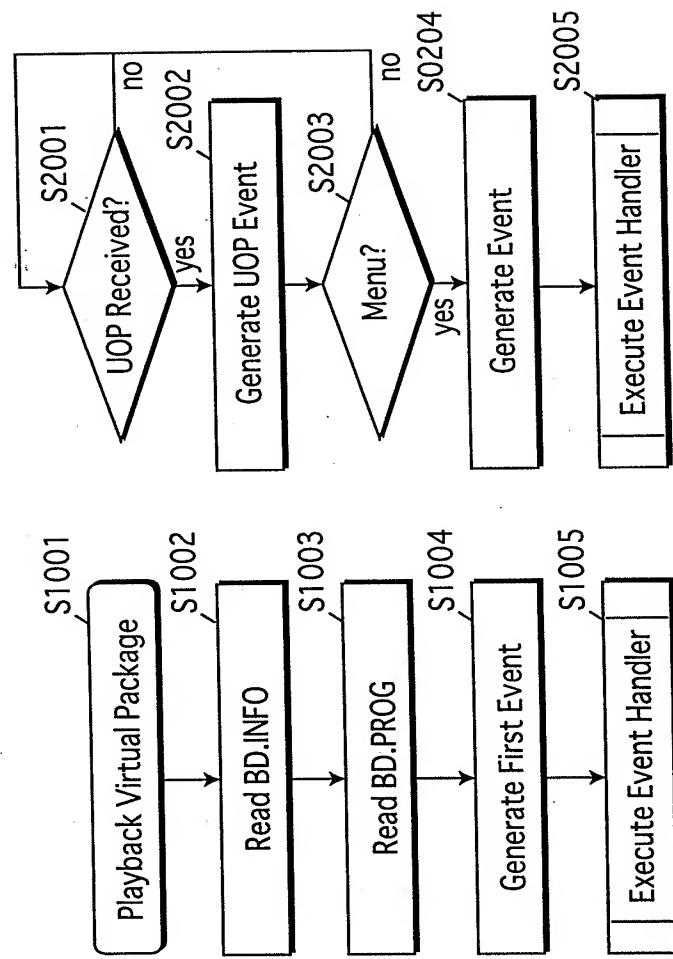


FIG.51

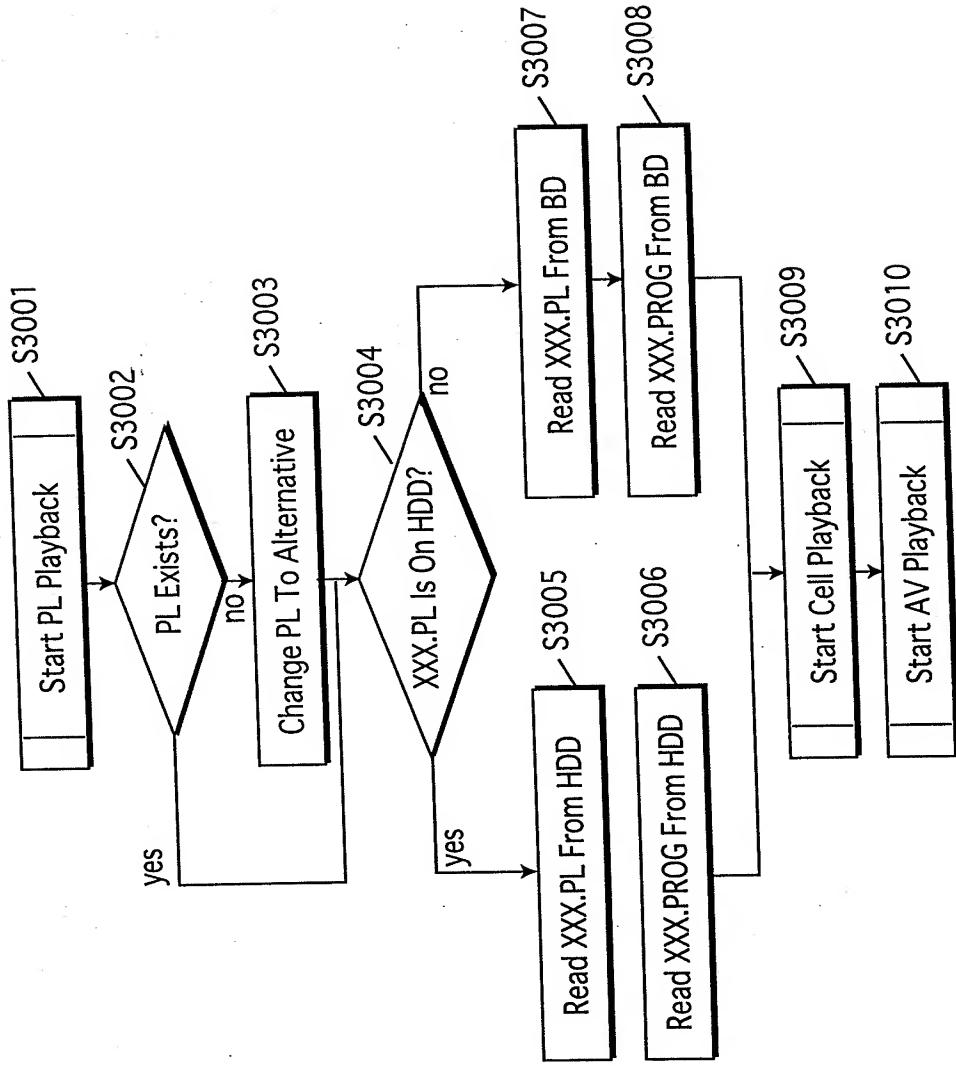


FIG.52

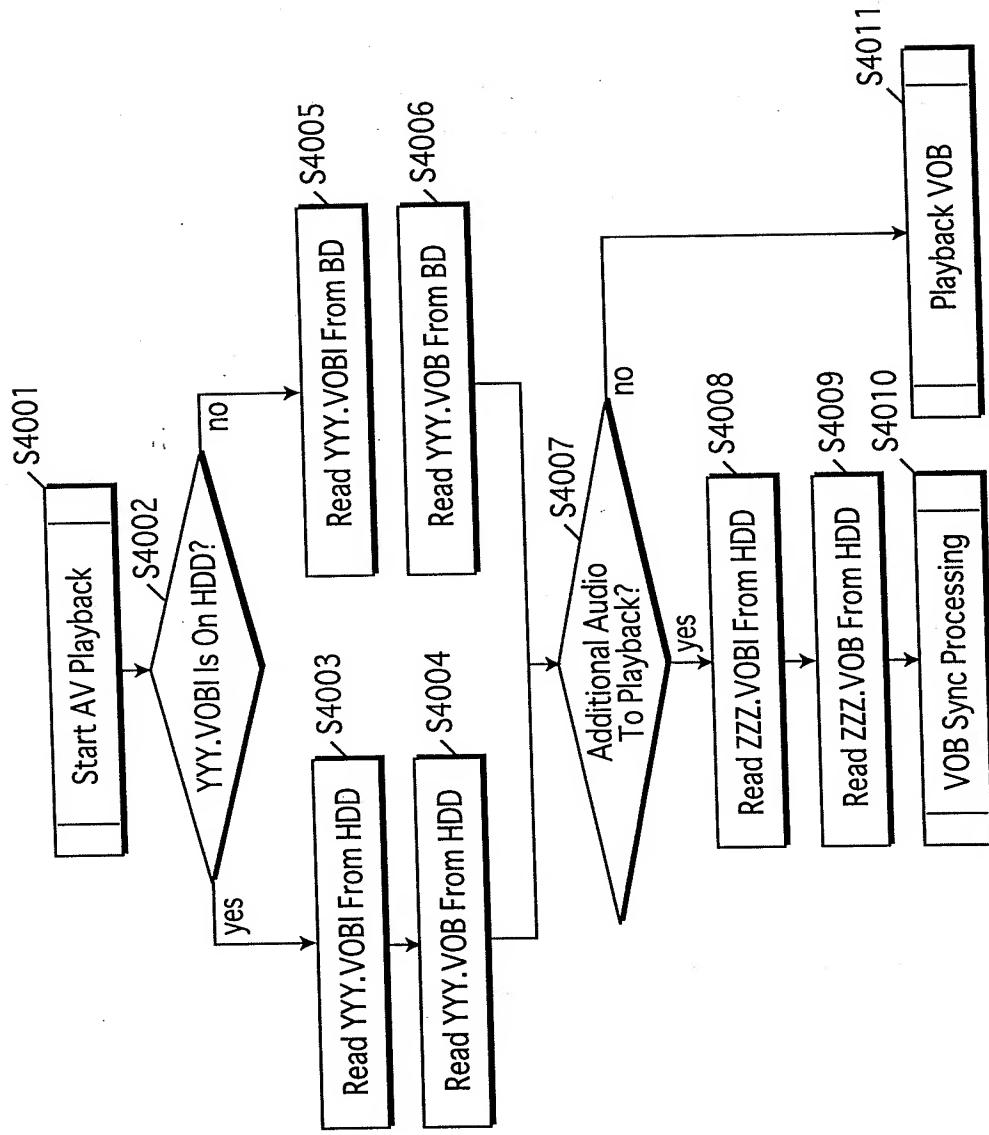


FIG.53

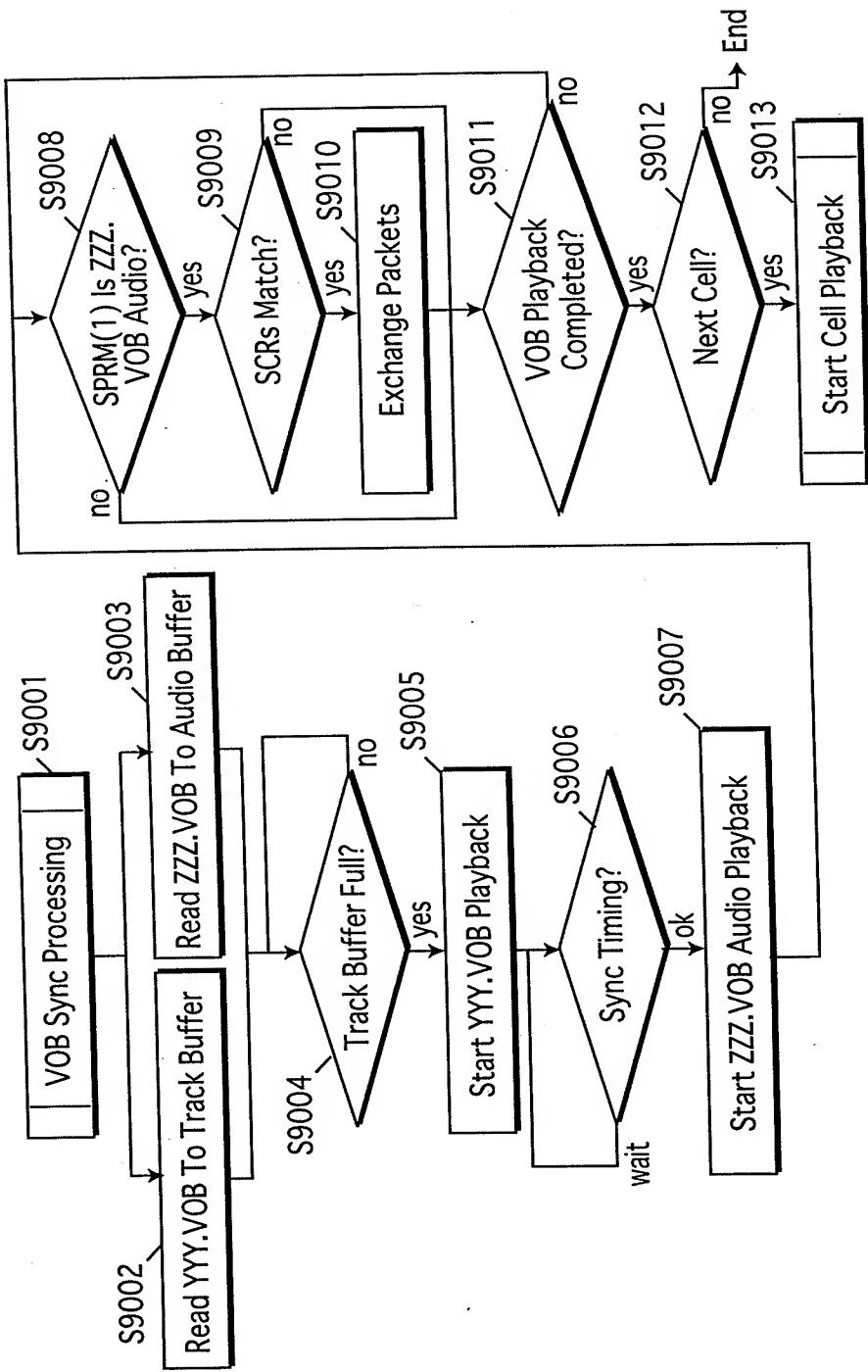


FIG.54

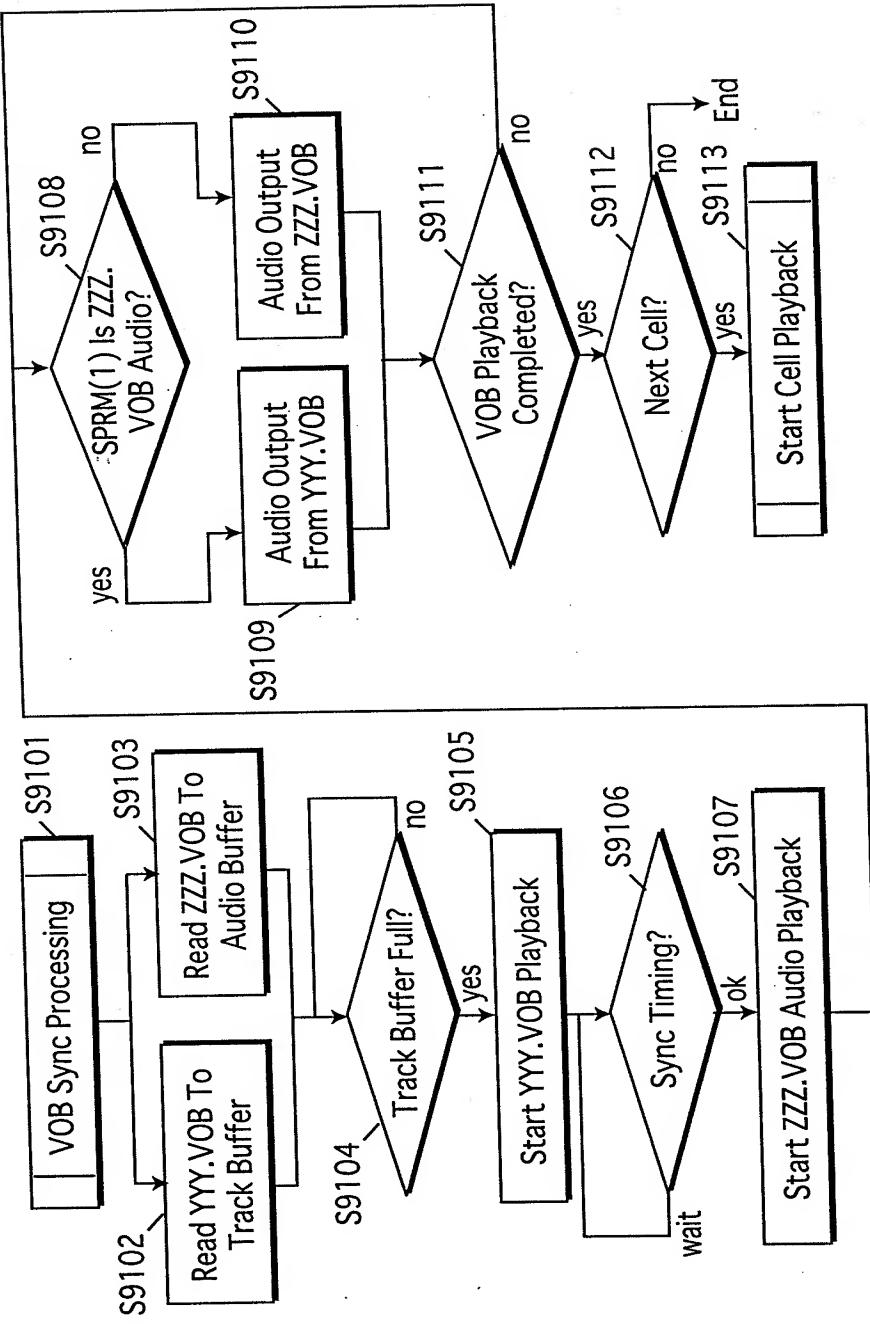


FIG.55

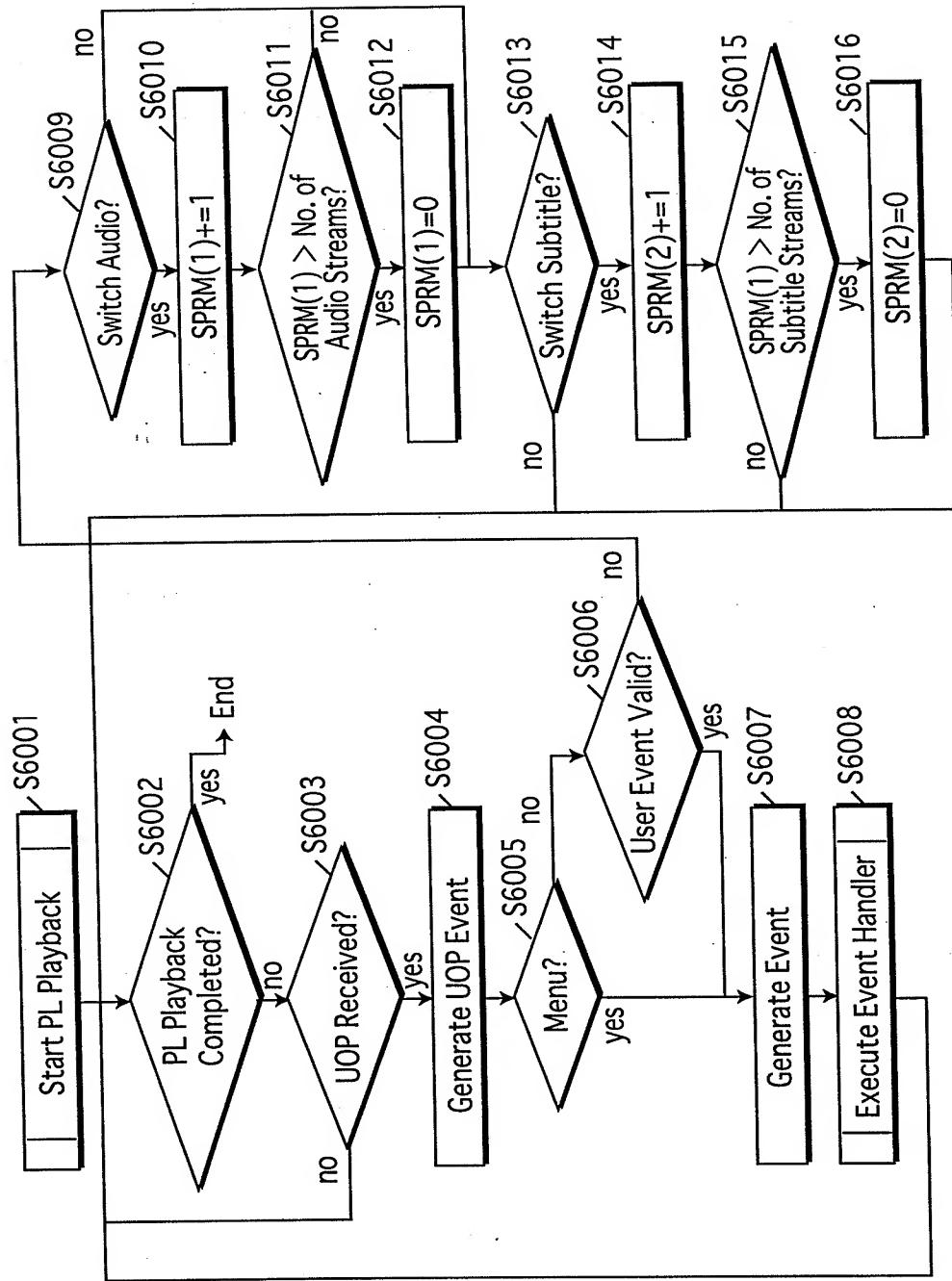


FIG.56

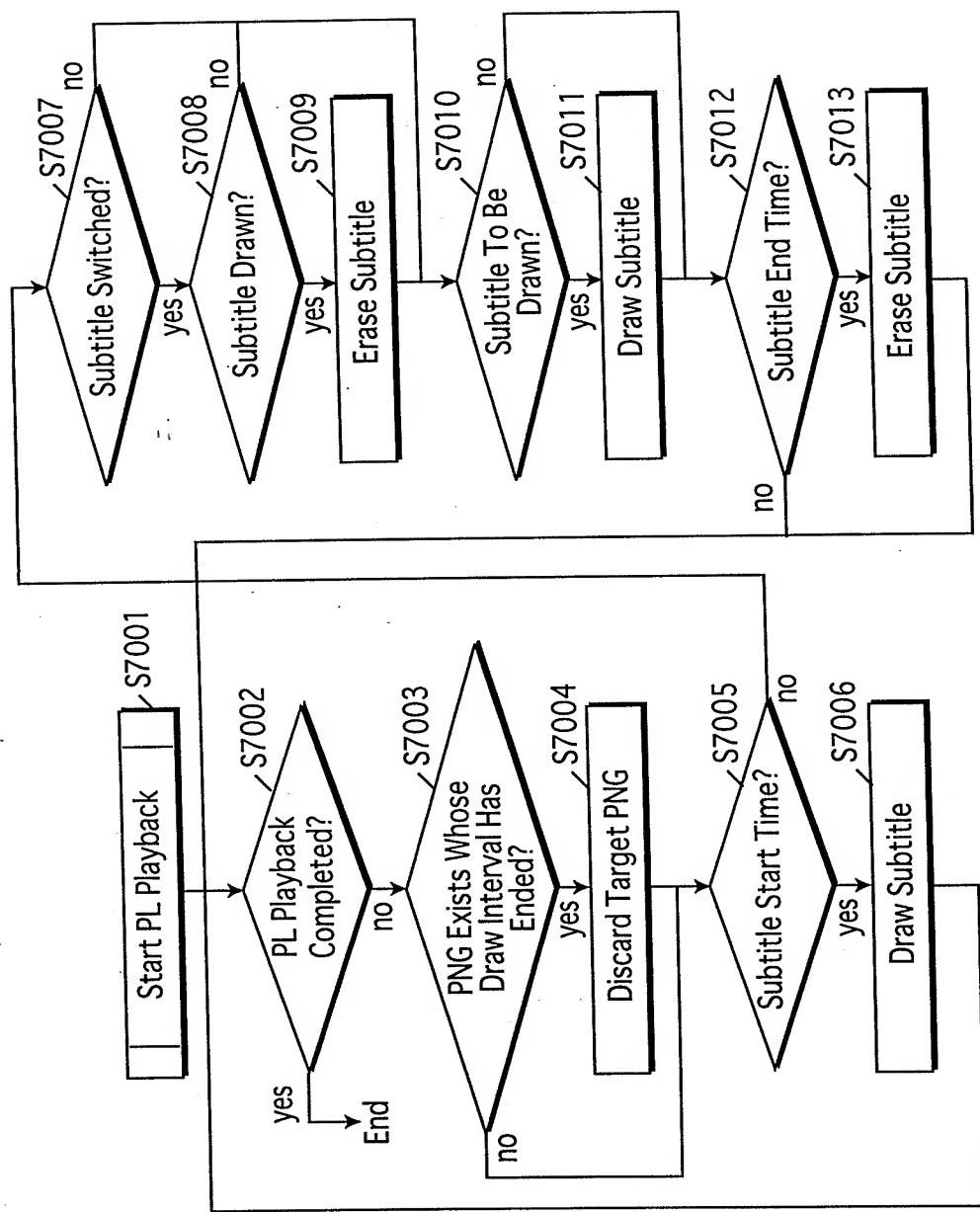


FIG.57

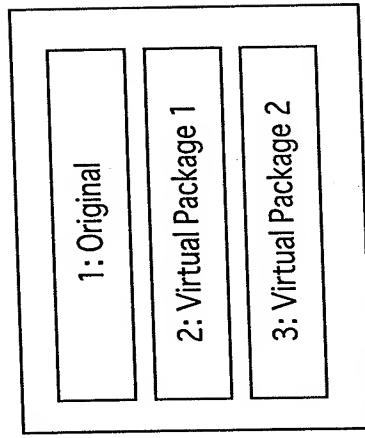


FIG.58

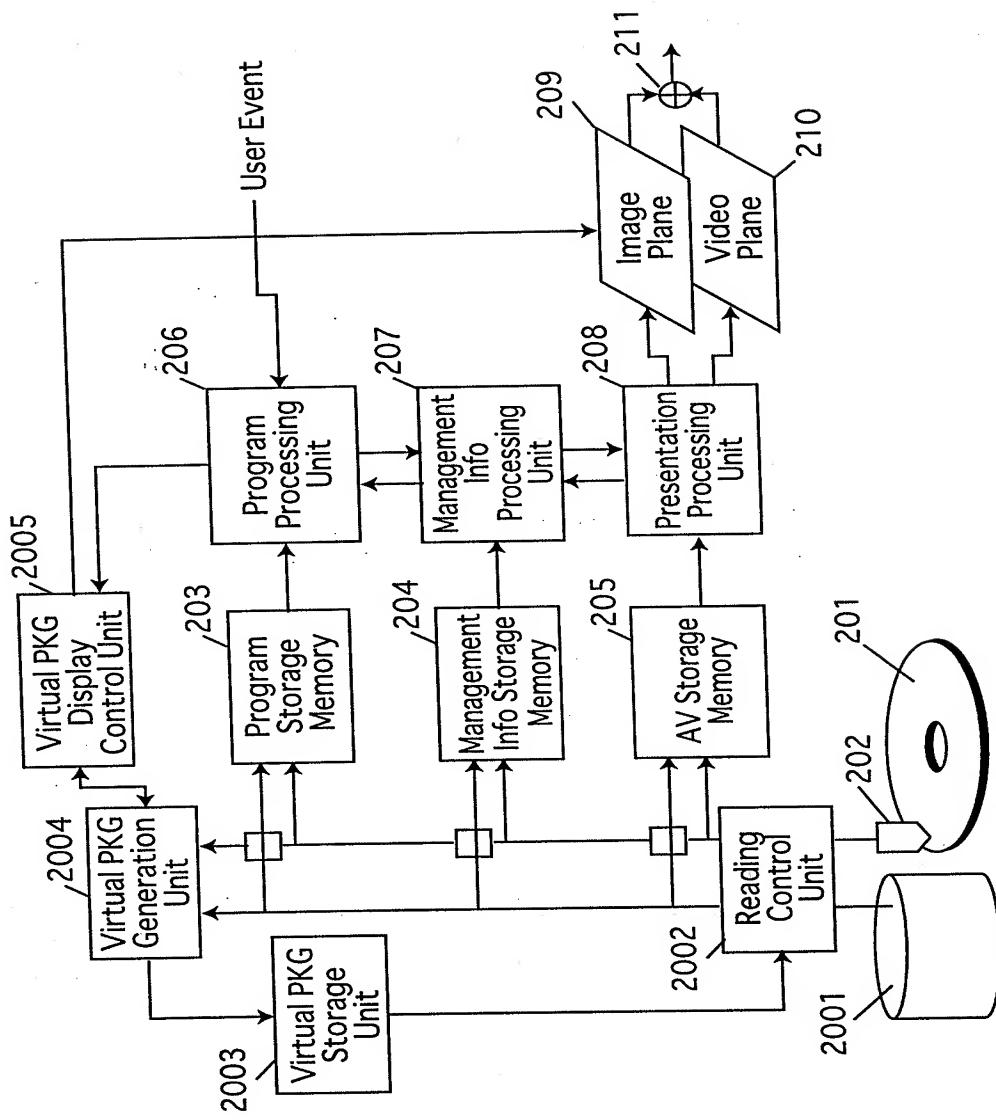


FIG.59

Player Variables (System Parameters)

0	Language Code	11	Player audio mixing mode for Karaoke	22	reserved
1	Audio stream number	12	Country code for parental management	23	Player status
2	Subtitle stream number	13	Parental level	24	Virtual package version
3	Angle number	14	Player configuration for Video	25	reserved
4	Title number	15	Player configuration for Audio	26	reserved
5	Chapter number	16	Language code for AST	27	reserved
6	Program number	17	Language code ext. for AST	28	reserved
7	Cell number	18	Language code for STST	29	reserved
8	Key name	19	Language coded ext. for STST	30	reserved
9	Navigation timer	20	Player region code	31	reserved
10	Current playback time	21	reserved	32	reserved

FIG.60

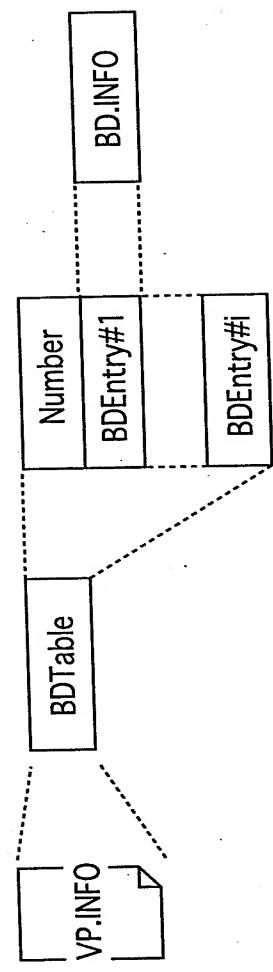


FIG.61

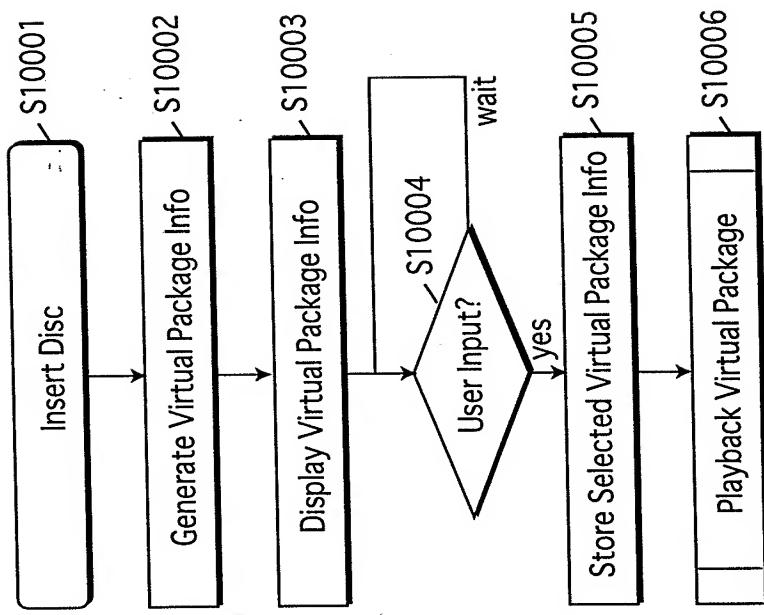


FIG.62

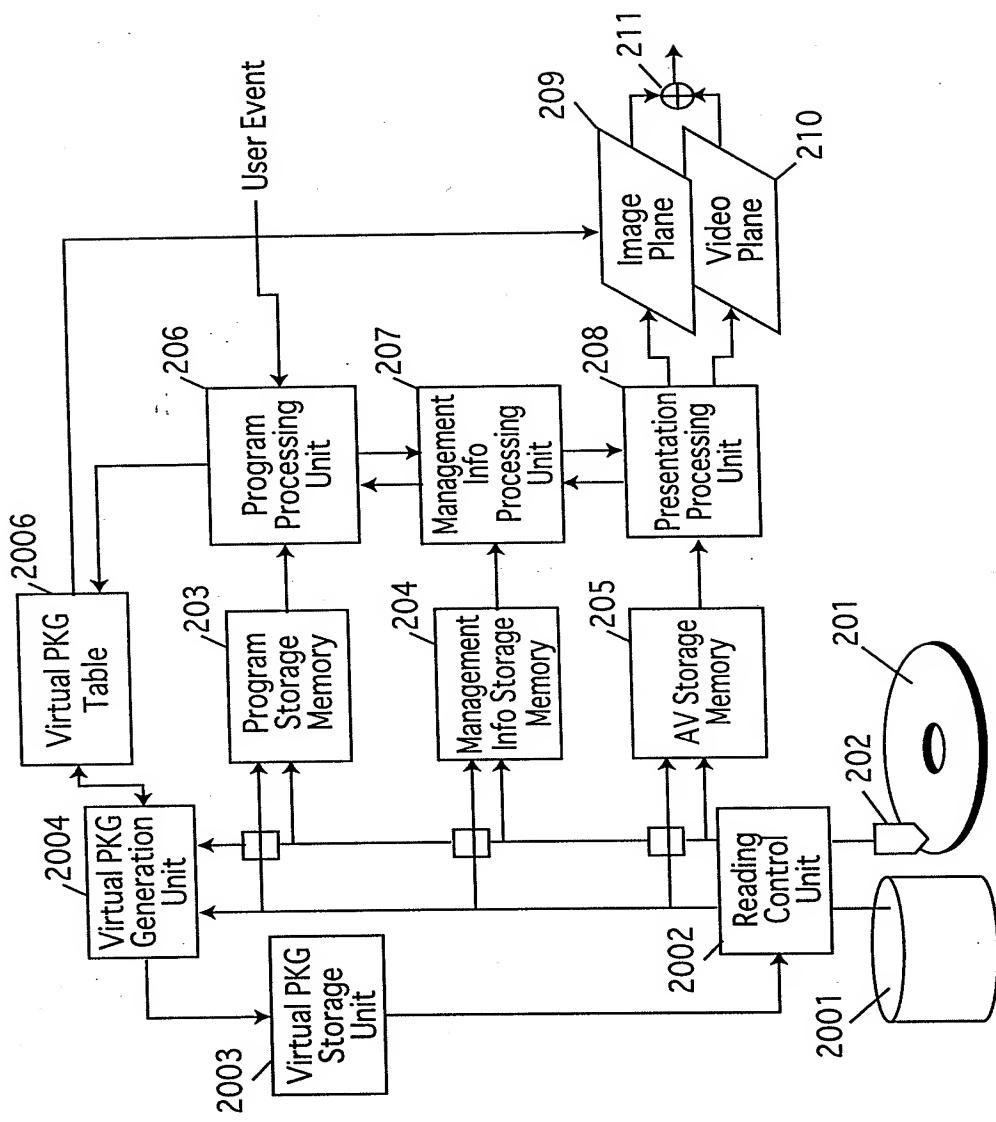


FIG.63

Package Name	Storage Position Information	End Position Information
ABC	¥ABC	PL#2/Cell#1/00:03:24:00
XYZ	-----	PL#1/Cell#1/00:13:05:10
WWW	Memory:¥WWW	-----
JKF	Server:¥JKF	PL#4/Cell#2/00:01:22:00

FIG.64

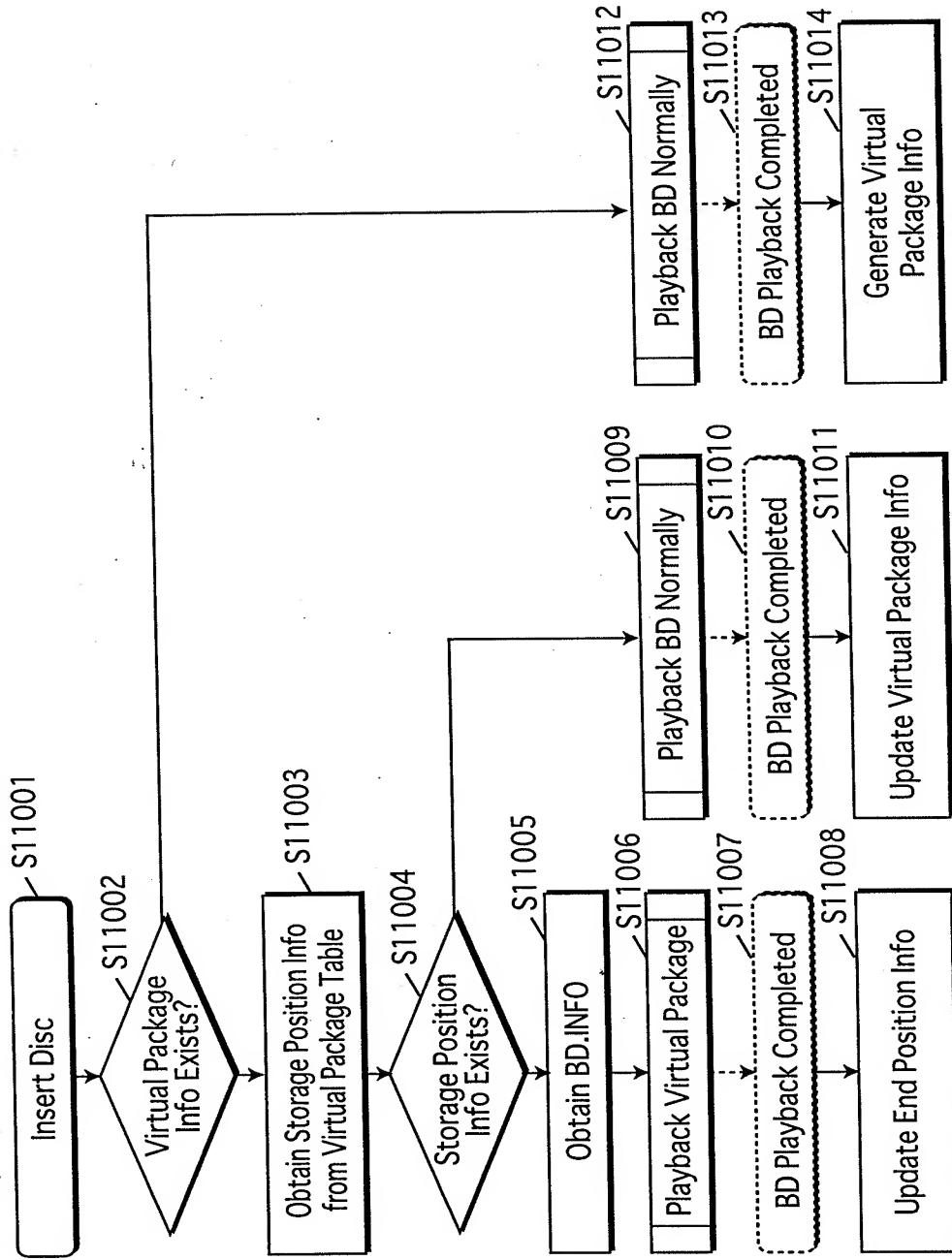


FIG.65

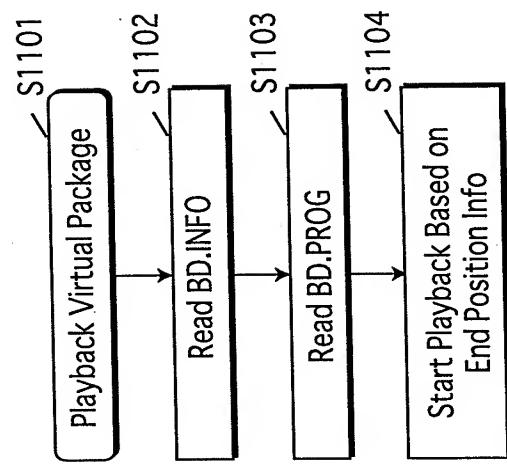


FIG.66

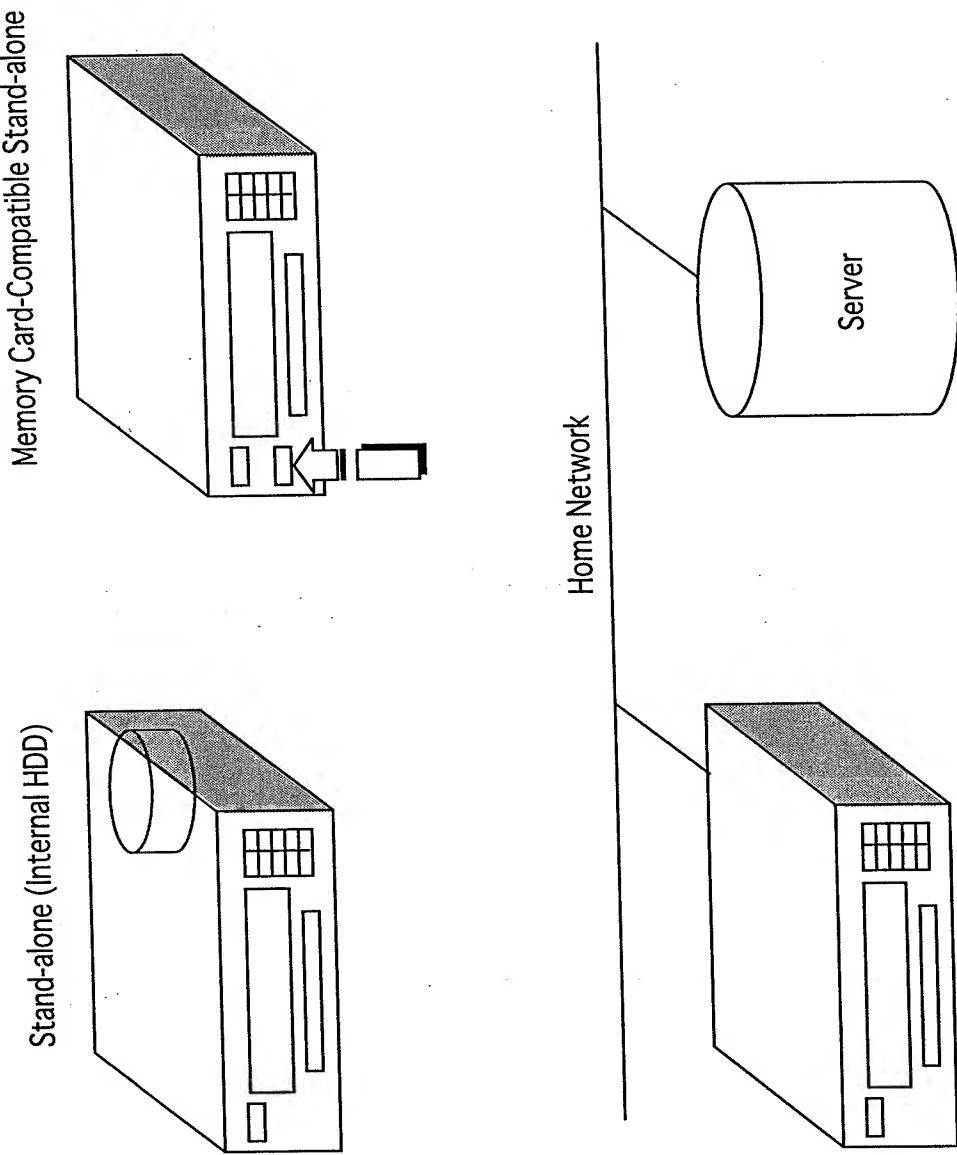


FIG.67

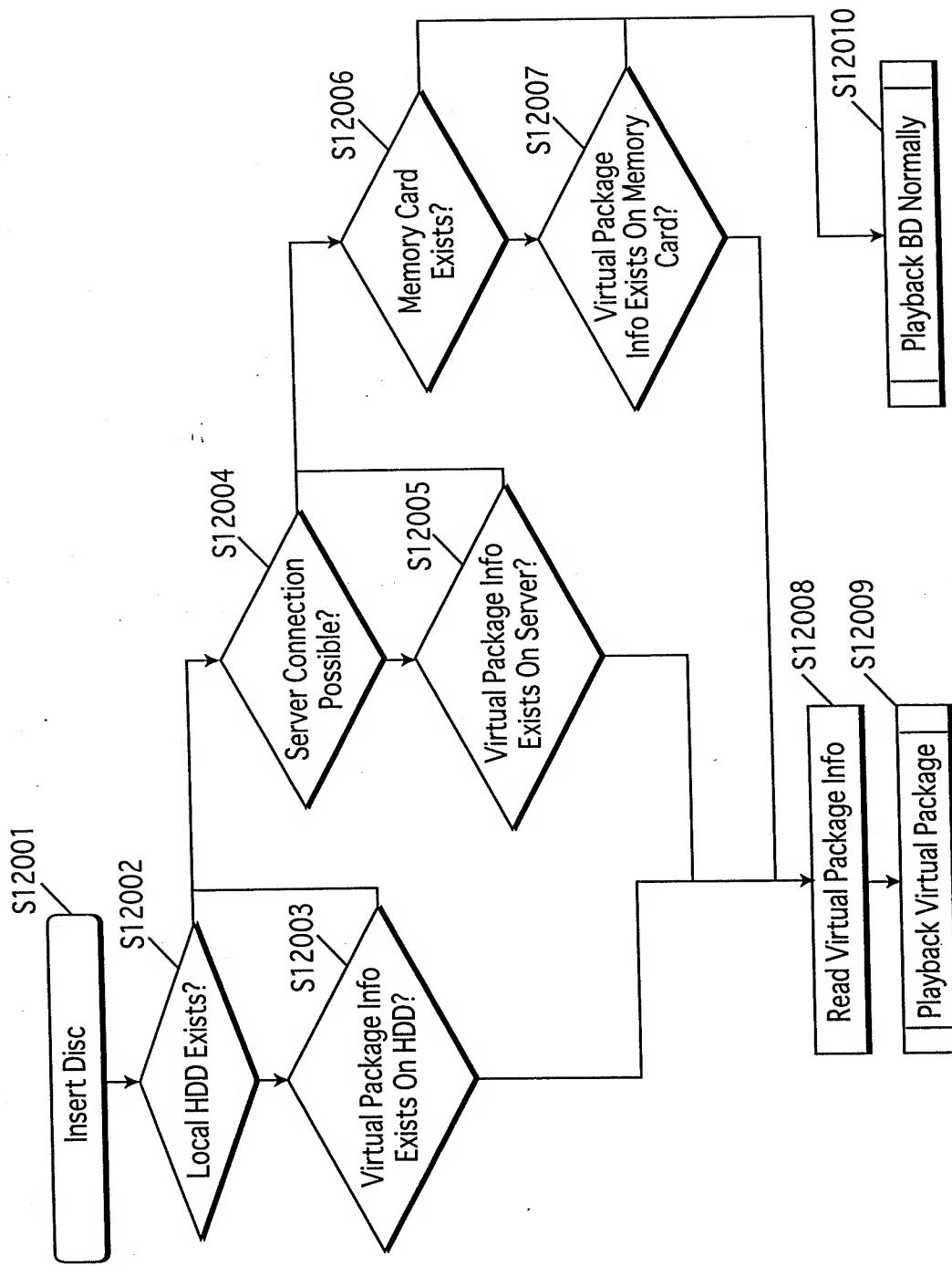


FIG.68

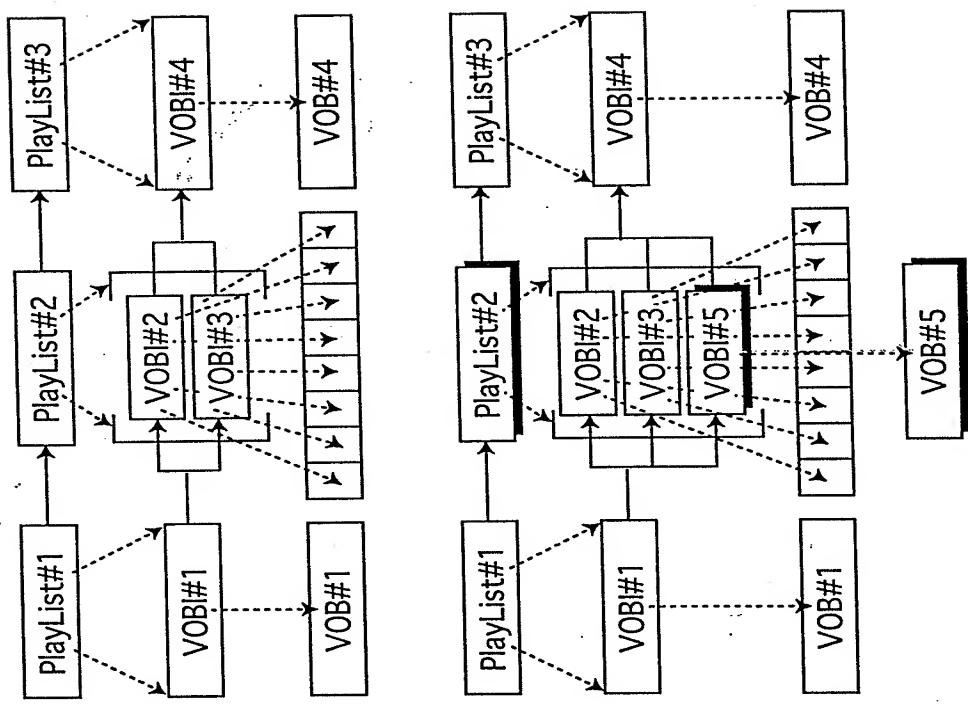


FIG.69

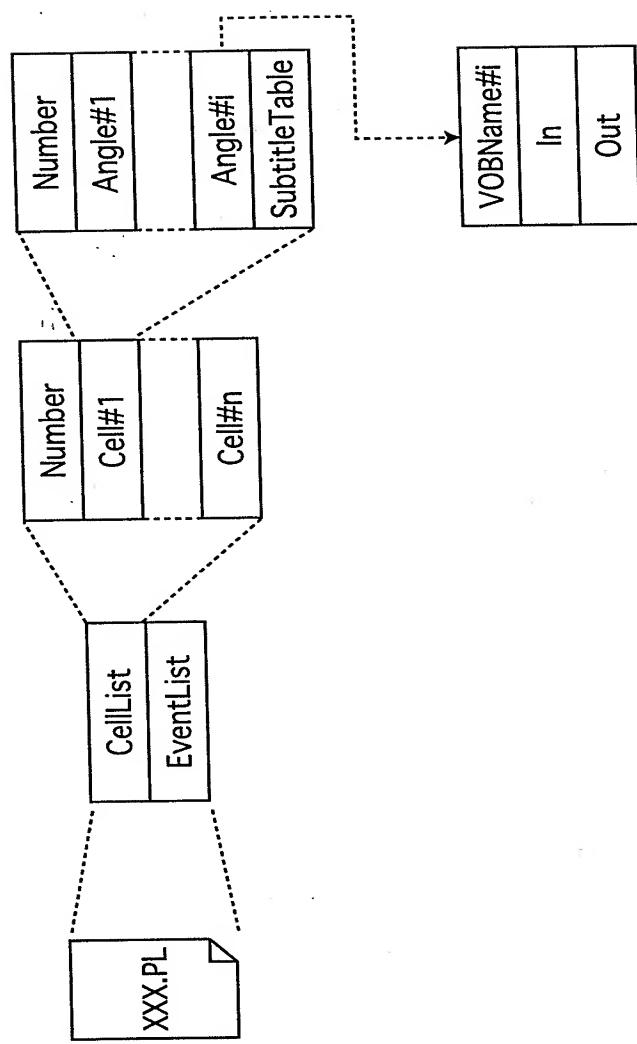


FIG.70

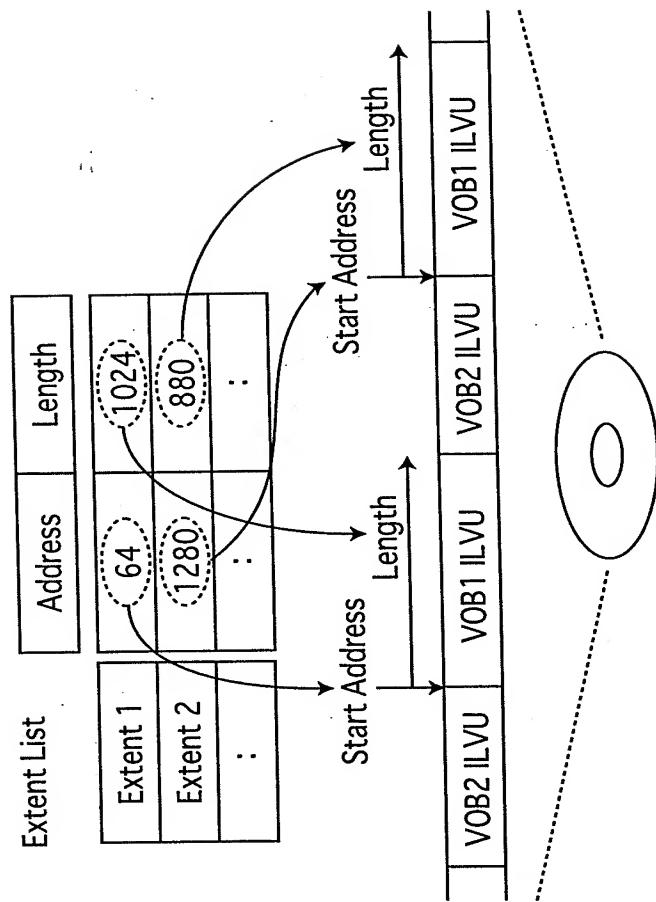


FIG.71

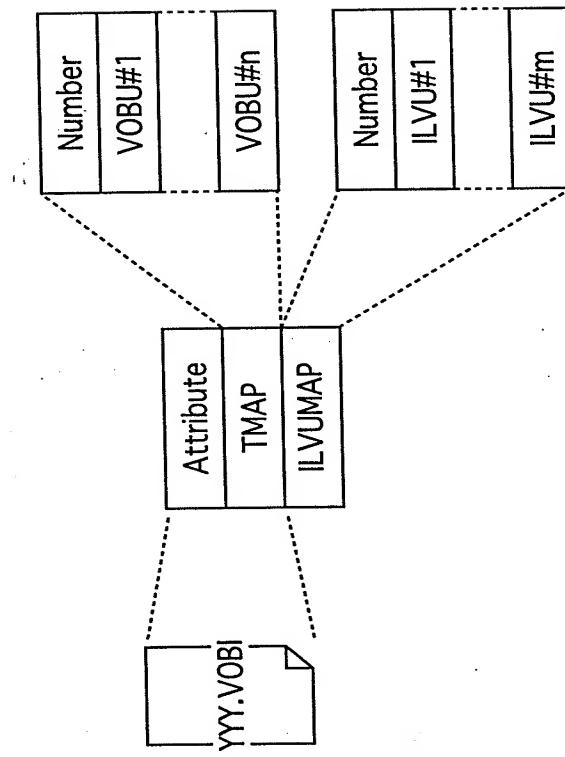


FIG.72

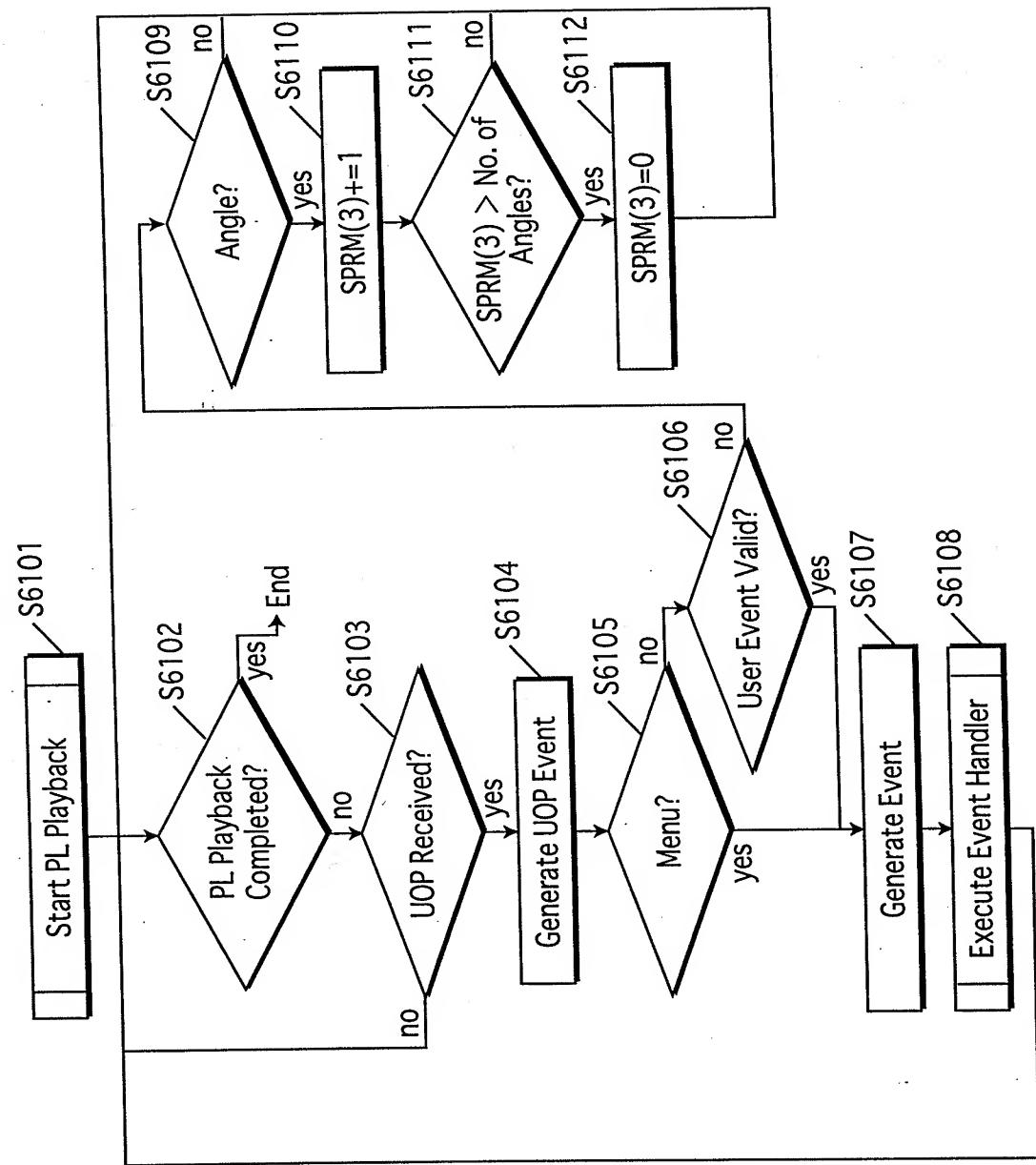


FIG.73

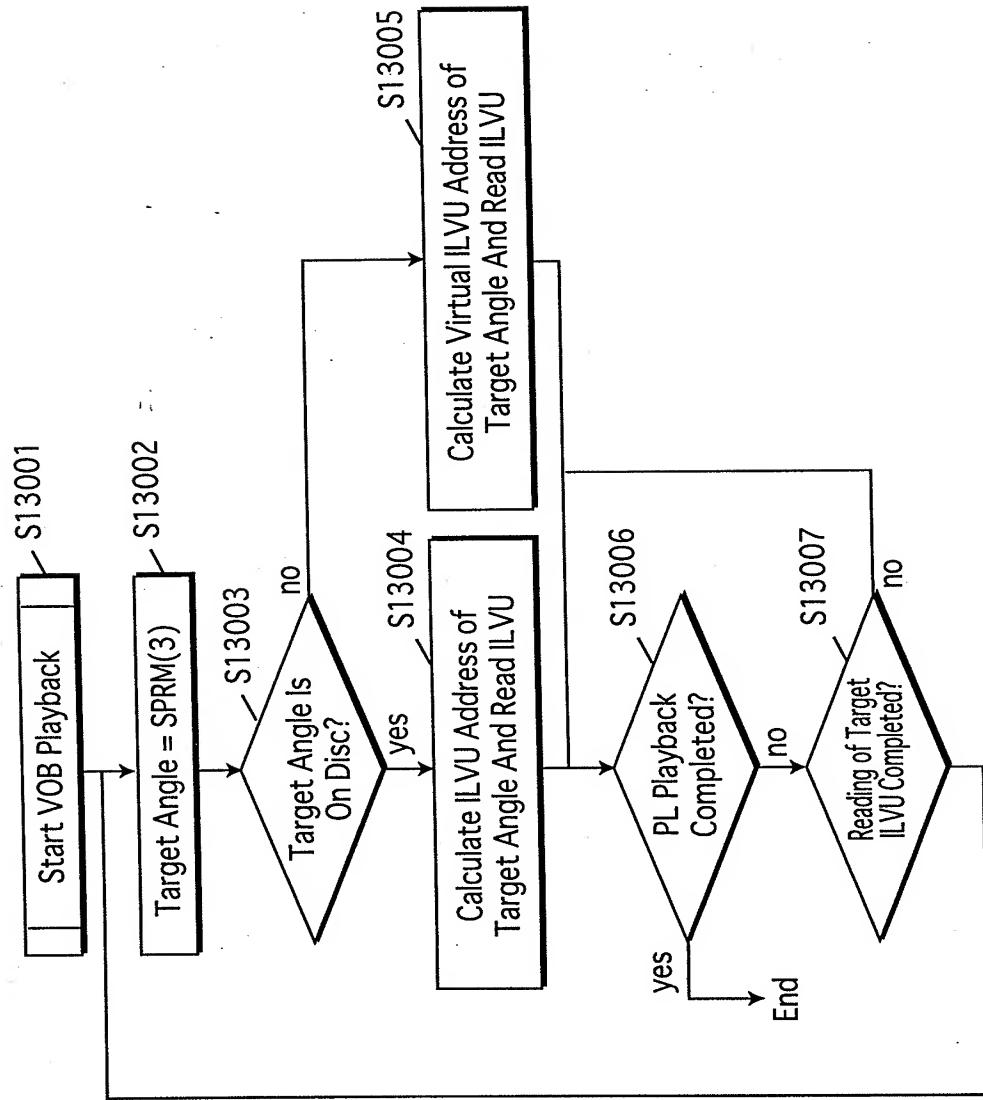


FIG. 74

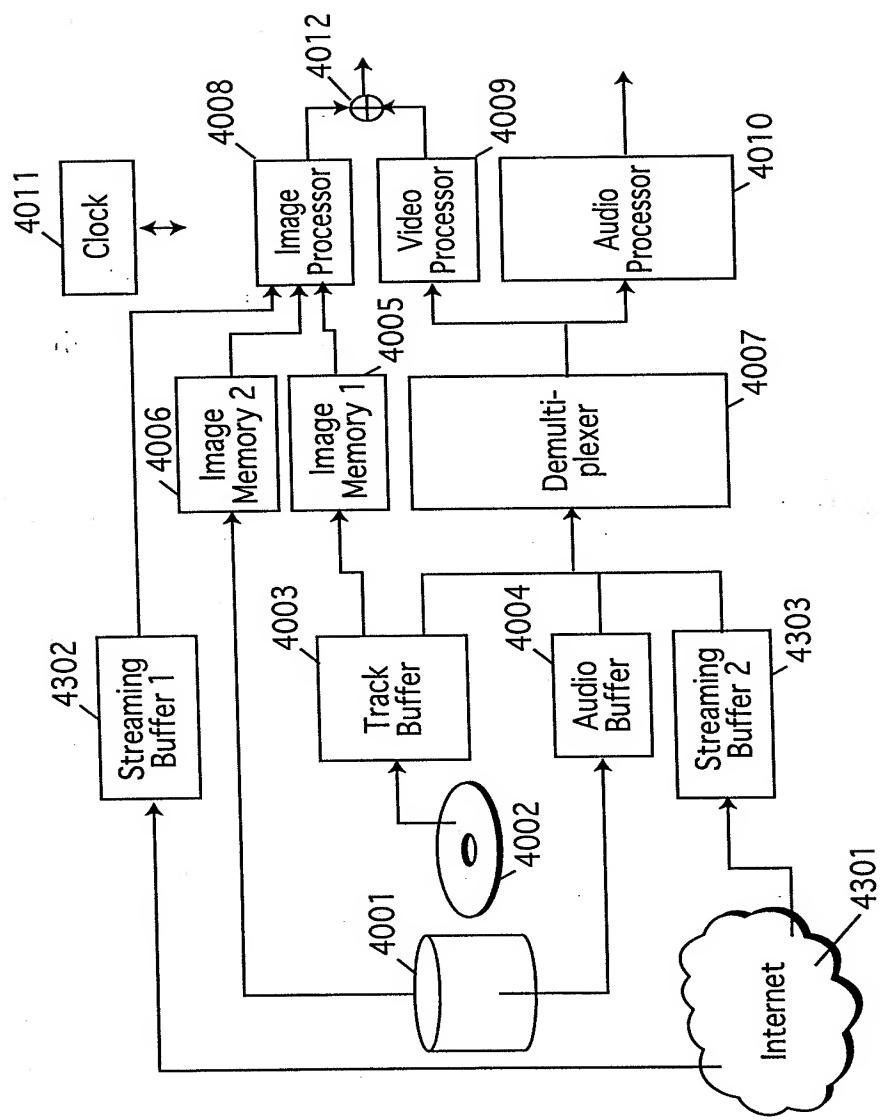


FIG.75

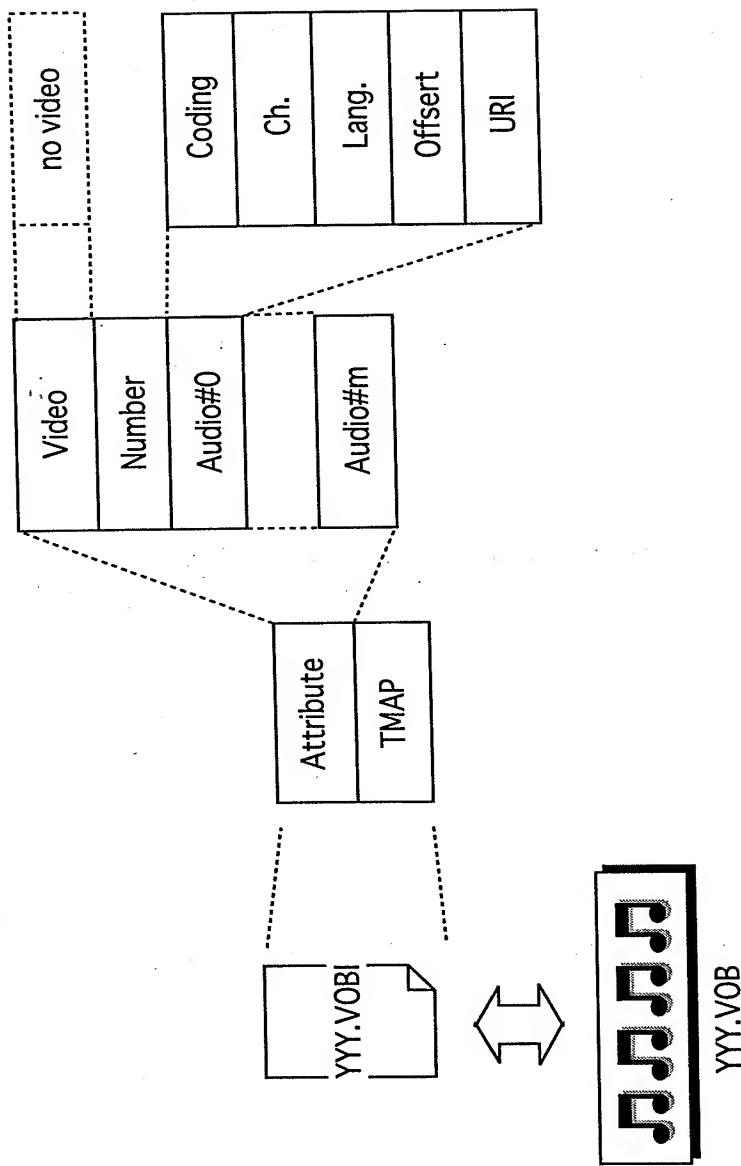


FIG.76

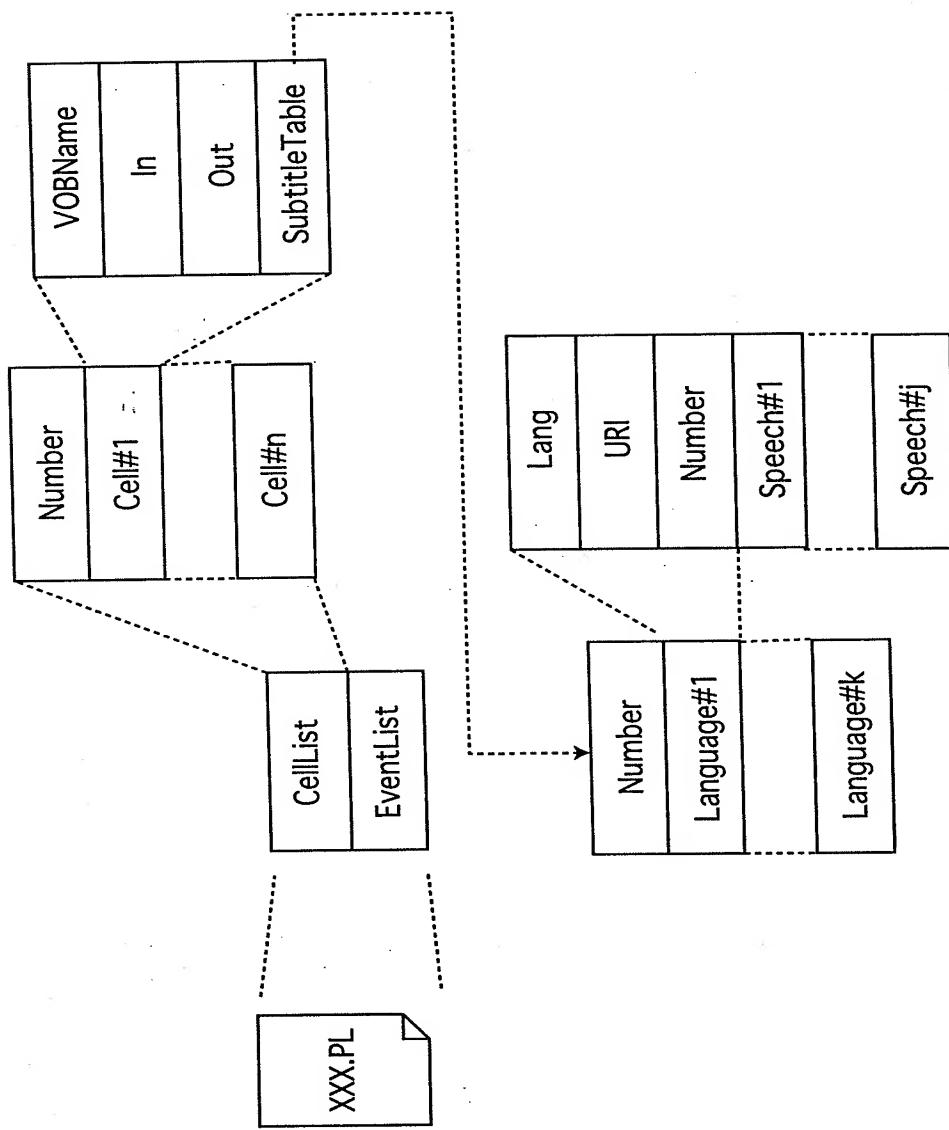


FIG.77

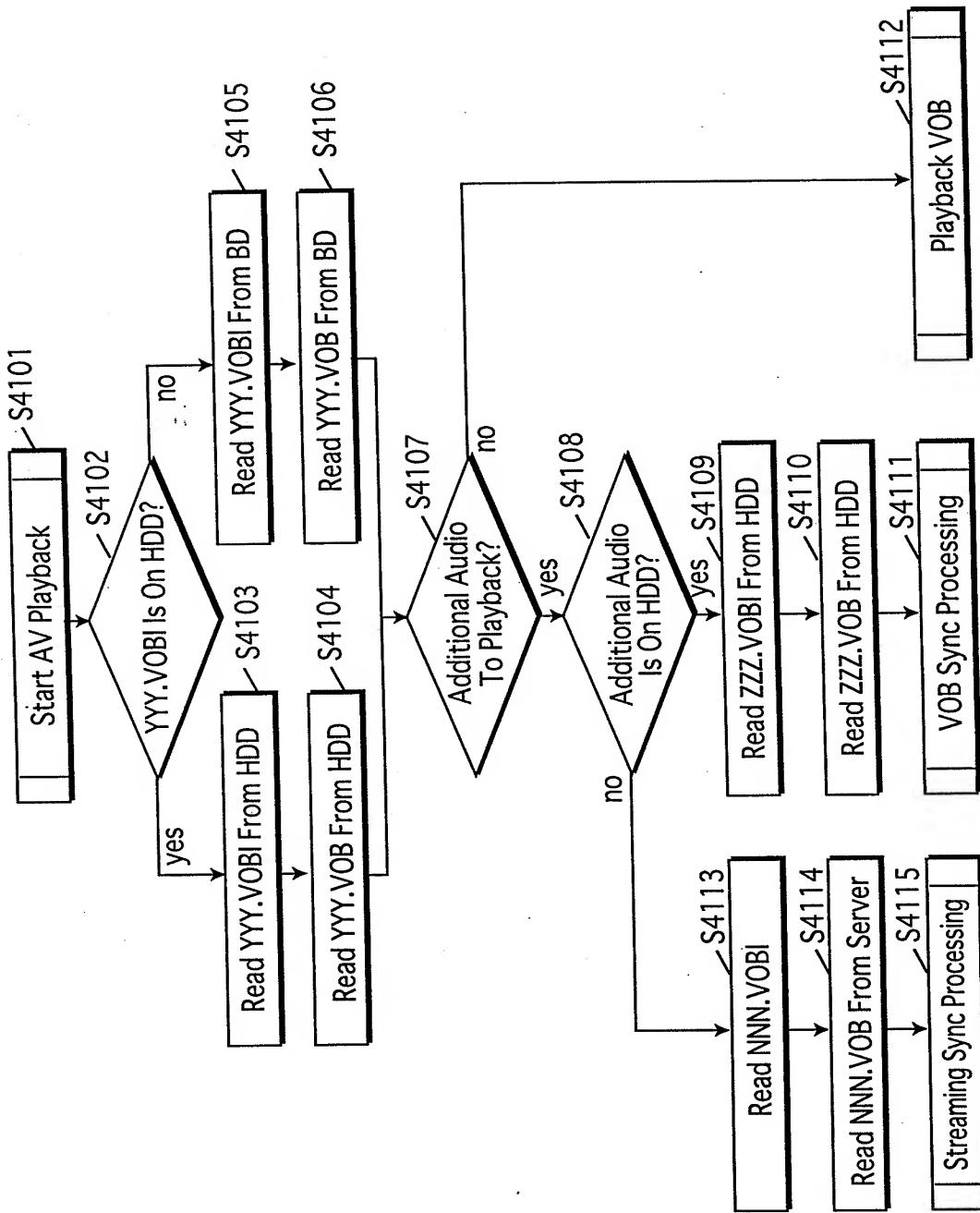


FIG.78

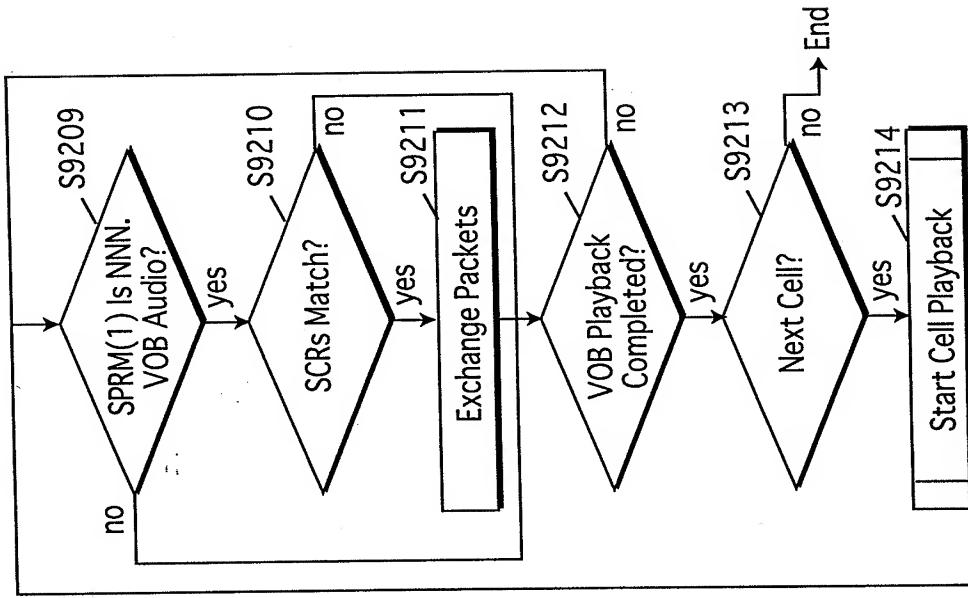
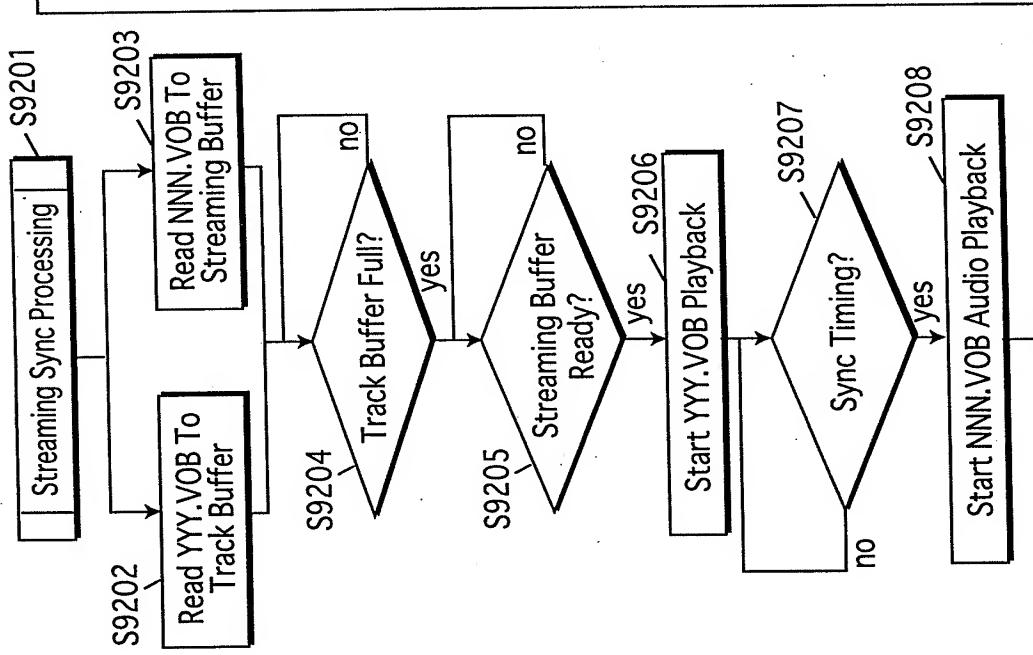


FIG.79

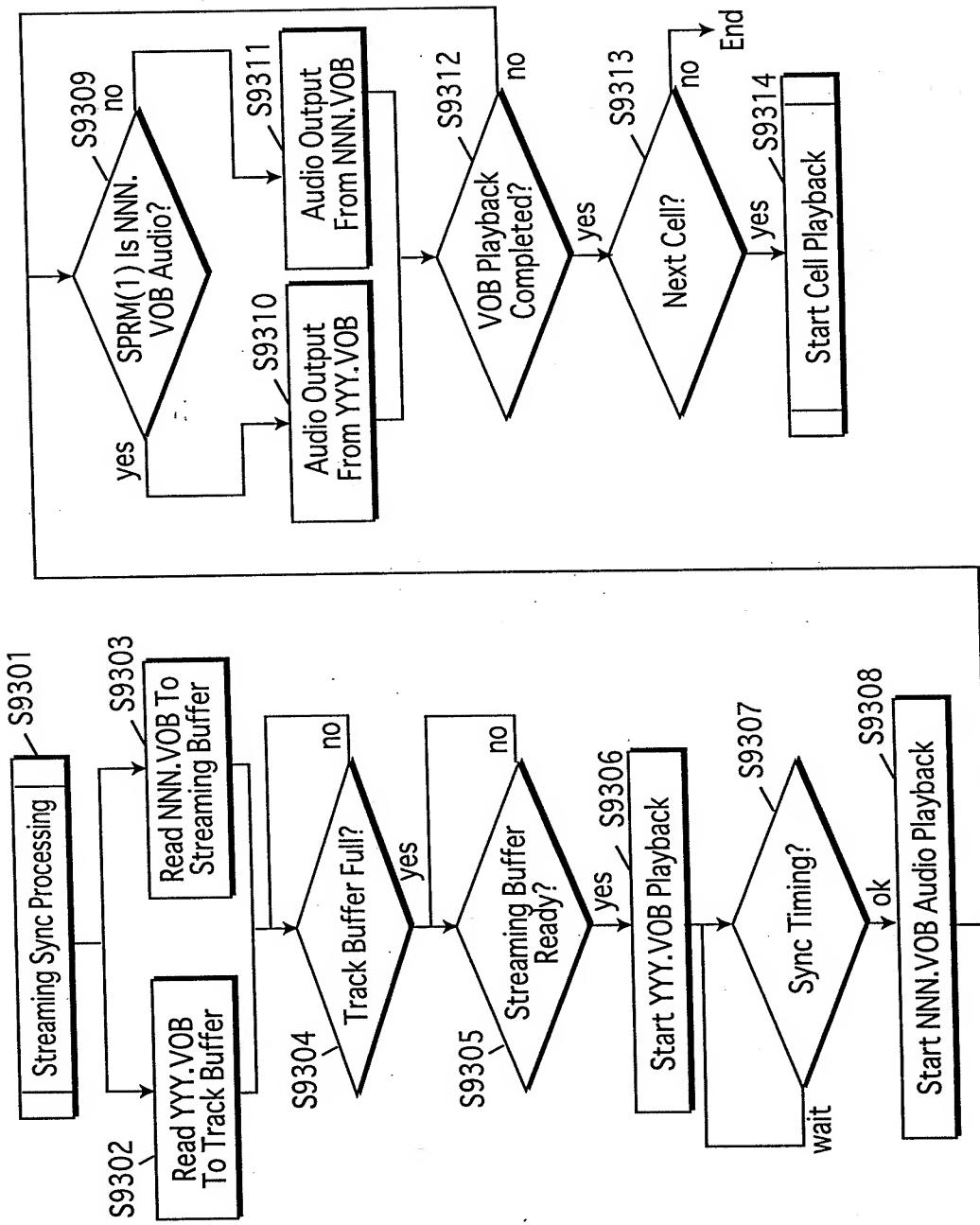


FIG.80

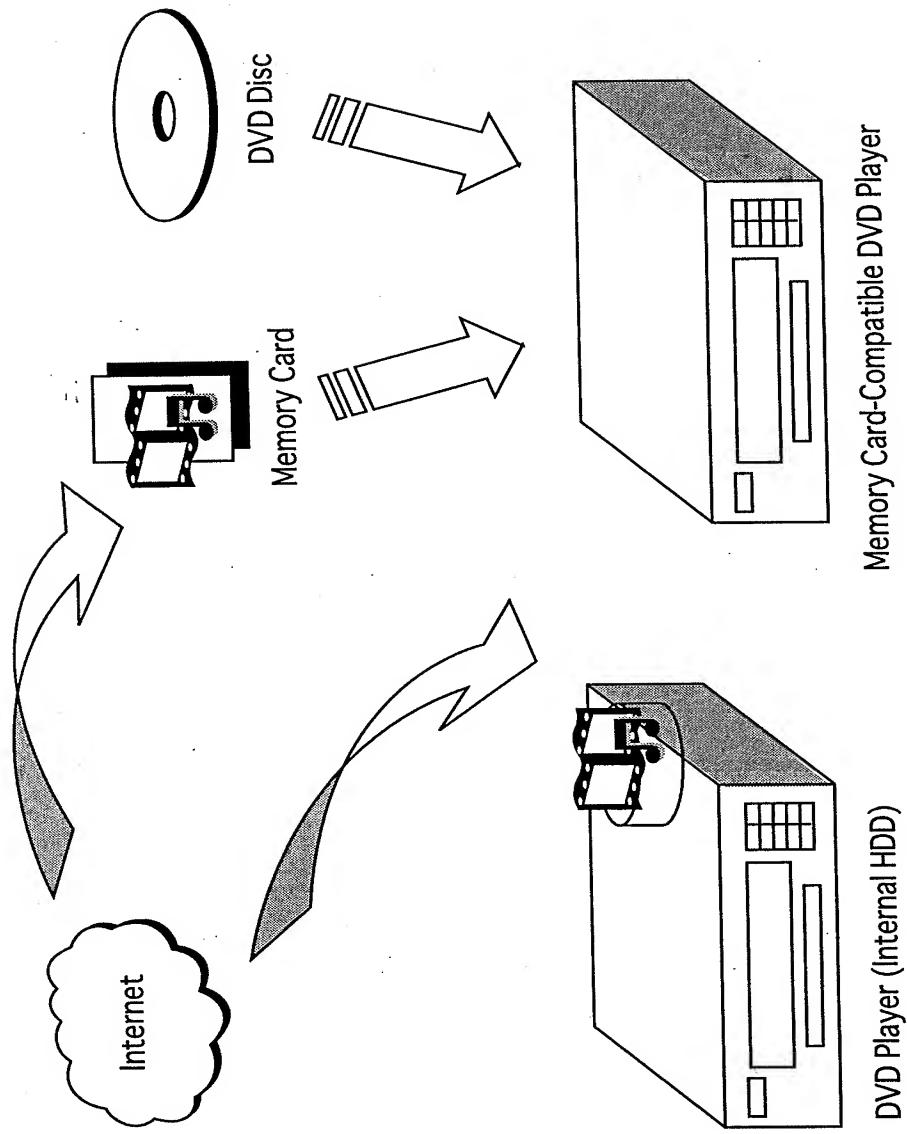


FIG.81

